



Graphisme sur le Web

Jean-Claude Moissinac

Jean-claude.moissinac@telecom-paristech.fr

Cyril Concolato

cyril.concolato@telecom-paristech.fr

2015



Web graphique?

■ Technologies de représentations graphiques utilisables pour le Web

- Principalement des standards du Web (hors de Flash)

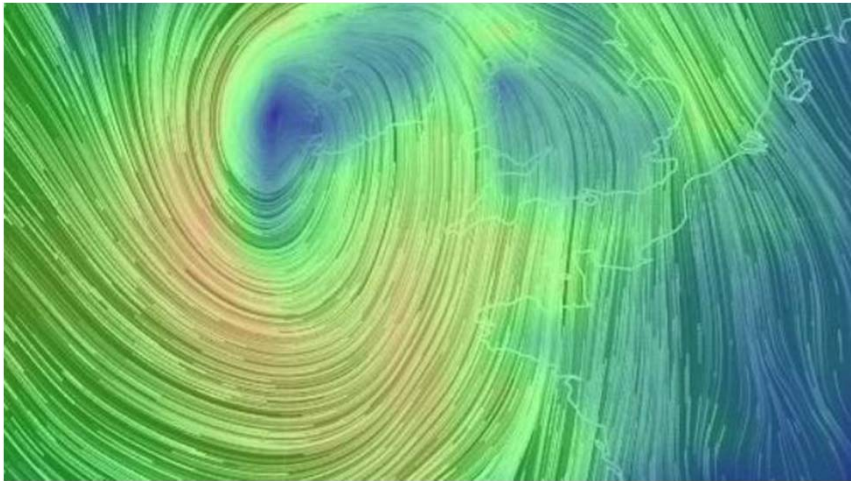
■ Utilisation

- Art, clipart
- Design
- Interfaces utilisateur
- Dessin animé, bande dessinée
- Publicités
- Sciences, présentation de données
- Cartographie
- Jeux
- Education
- Multimedia
- ...



Joshua Davis

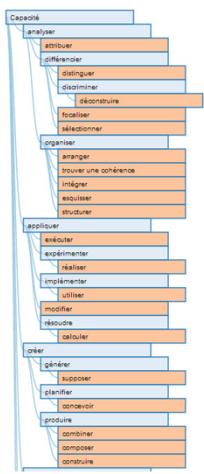
Demo 1



page 2

TELECOM ParisTech


Demo 2



page 3

TELECOM ParisTech

Demo 3



A cartoon illustration of a young boy with a red cap, green t-shirt, and blue pants, holding a large orange basketball. He is standing with one hand on his hip and the other holding the ball.

page 4


TELECOM
ParisTech

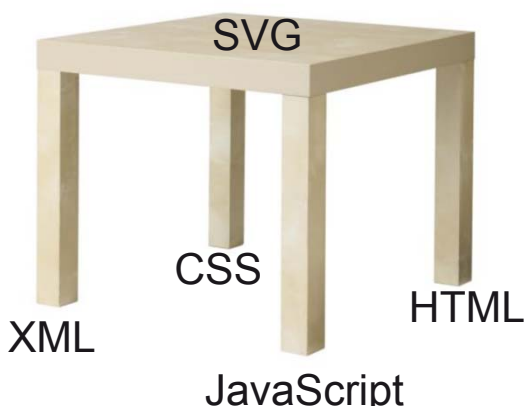
Aperçu


- **Rappels**
- **Les technologies abordées**
 - SVG
 - CSS
 - SVG & CSS
 - SVG & HTML
 - SVG & JavaScript
 - Canvas
 - Canvas 2D
 - WebGL

page 5

TELECOM
ParisTech

 **Rappels**



page 6 

 **XML et multimédia**



Qu'ont-ils en commun?











$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Extrait de la feuille suivante sur MathML

TELECOM ParisTech

Qu'ont-ils en commun?

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Extrait de la feuille suivante sur MathML


MusicML

SVG

MathML

MMS/SMIL

TELECOM ParisTech




XHTML

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Java 1&grave;re ann&eacute;e</title>
<meta name="generator" content="S5" />
<meta name="version" content="S5 1.1" />
...
<!-- style sheet links -->
<link rel="stylesheet"
href="ui/default/slides.css" type="text/css"
media="projection" id="slideProj" />
...
<!-- S5 JS -->
<script src="ui/default/slides.js"
type="text/javascript"></script>
</head>
<body>
<div class="layout">
<div id="controls">
<div id="currentSlide">
<div id="header">
<div id="footer">
<h1>Telecom ParisTech</h1>
<h2>Programmation O.O. en Java</h2>
</div>

```




MusicML

```

<score-partwise version="3.0">
<part-list>
<score-part id="P1">
<part-name>Music</part-name>
</score-part>
</part-list>
<part id="P1">
<measure number="1">
<attributes>
<divisions>1</divisions>
<key>
<fifths>0</fifths>
</key>
<time>
<beats>4</beats>
<beat-type>4</beat-type>
</time>
<clef>
<sign>G</sign>
<line>2</line>

```




MMS/SMIL

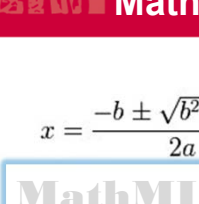
MMS/SMIL

```

<smil>
  <head>
    <meta name="title" content="vacation photos" />
    <meta name="author" content="Danny Wyatt" />
  </head>
  <layout>
    <root-layout width="160" height="120"/>
    <region id="Image" width="100%"
      height="80" left="0" top="0" />
    <region id="Text" width="100%"
      height="40" left="0" top="80" />
  </layout>
  <body>
    <par dur="8s">
      
      <text src="FirstText.txt" region="Text" />
      <audio src="FirstSound.amr"/>
    </par>
    <par dur="7s">
      
      <text src="SecondText.txt" region="Text" />
      <audio src="SecondSound.amr" />
    </par>
  </body>
</smil>

```





MathML

MathML


$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```

<math>
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mo>-</mo>
        <mi>b</mi>
        <mo>+</mo>
        <msqrt>
          <mrow>
            <msup>
              <mi>b</mi>
              <mn>2</mn>
            </msup>
            <mo>-</mo>
            <mrow>
              <mn>4</mn>
              <mi>a</mi>
              <mi>c</mi>
            </mrow>
          </mrow>
        </msqrt>
      </mrow>
      <mrow>
        <mn>2</mn>
        <mi>a</mi>
      </mrow>
    </mfrac>
  </mrow>
</math>

```


SVG




```
<?xml version='1.0' encoding='iso-8859-1'?>
<svg width='100%' height='100%' xmlns='http://www.w3.org/2000/svg'
xmlns:xlink='http://www.w3.org/1999/xlink' viewBox='0 0 1600 1200'>
<g id="s7"> <g fill='black' display='inline' fill-rule='evenodd' stroke='none'
<path id='C1582' fill='#000000' opacity='1.000000' display='inline' d='M42
<path id='C1572' fill='#Dadbe0' opacity='1.000000' d='M382 105711 011 011
<path id='C1570' fill='#Ffbd94' opacity='1.000000' d='M394 67511 010 -110 -
<path id='C1514' fill='#708f70' opacity='1.000000' d='M389 62311 010 -111 0
...
</g>
</svg>
```

TELECOM ParisTech

page 14


Bonus

RSS



```
<rss version="2.0">
<channel>
<title>Mon site</title>
<description>Exemple de flux RSS
2.0</description>
<lastBuildDate>Sat, 07 Sep 2002
00:00:01 GMT</lastBuildDate>
<link>http://www.example.org</link>
<item>
<title>Actualité N°1</title>
<description>Ceci est ma première
actualité</description> <pubDate>Sat, 07
Sep 2002 00:00:01 GMT</pubDate>
<link>http://www.example.org/actu1</link>
</item></channel> </rss>
```

KML



```
<kml xmlns="http://earth.google.com/kml/2.2">
<Document>
<name>Consignes Atterrissage Saint-Nazaire-Les-Eymes</name>
<description><![CDATA[Atterrissage de Saint-Nazaire-Les-Eymes
Consignes impératives à respecter pour la pérennité du
site]]></description>
<Style id="style2">
<LineStyle>
<color>73FF0000</color>
<width>5</width>
</LineStyle>
</Style>
<Placemark>
<name>Lignes électrique</name>
<description><![CDATA[]]></description>
<styleUrl="#style2"/>
<ExtendedData>
<Data name="_SnapToRoads">
<value>true</value>
</Data>
</ExtendedData>
<LineString>
<tessellate>1</tessellate>
<coordinates>
5.841270,45.265209,0.000000
5.840580,45.265759,0.000000
5.840580,45.265759,0.000000
5.840750,45.266411,0.000000
</coordinates>
</LineString>
```

TELECOM ParisTech

page 15



Les langages d'interface

- UIML
- XUL -> Mozilla/Firefox
- XAML -> Microsoft
- XML Layouts pour Android

page 16




XML LayoutsAndroid

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
  <TextView android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello, I am a TextView" />
  <Button android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello, I am a Button" />
</LinearLayout>
```

page 17



 **XAML**

```

<windows Width="600" Height="480" Text="Mon Programme">
  <FlowPanel>
    <Label Name="Montexte" FontSize="20"> Mon application </label>
    <Button Width="80" Click="BoutonClic"> Fermer </Button>
  </FlowPanel>
</windows>

```

```


<Canvas ID="root" xmlns="http://schemas.microsoft.com/2003/xaml"
  xmlns:def="Definition">
  <TextBlock>Hello World!</TextBlock>
</Canvas>


```

page 18

direction ou services

<pied de page>



 **XUL**

```


<window
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <box>
    <description>Hello world!</description>
  </box>
</window>

```

page 19

direction ou services

<pied de page>





Domaines nominaux: namespaces

- Espaces de noms
- Permet de préciser à quel type se réfère un élément
- But: éviter les collisions de noms, lorsqu'un document est composé d'éléments de plusieurs grammaires

■ Exemple:

```
<monelement xmlns:xmlamoi="http://xmlamoi.org/DTDProjet/1.0">
  <p>un paragraphe ... de l'espace de noms par défaut</p>
  <xmlamoi:p>un paragraphe ... de l'espace de noms xmlamoi</p>
</monelement>
```

- <http://www.w3.org/TR/REC-xml-names/>




W3C SVG Scalable Vector Graphics

Aperçu




The World Wide Web Consortium (W3C)

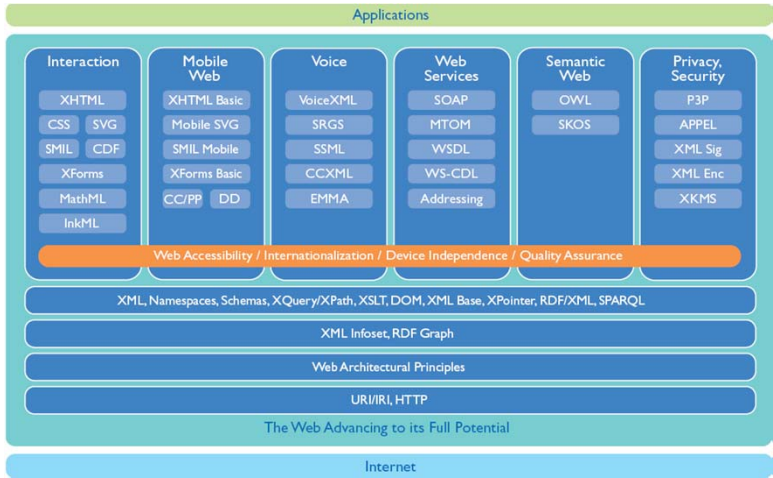
- <http://www.w3.org/>
- **Consortium international**
 - 300-400 membres
 - Créé en 1994, par Tim Berners-Lee
- **Produit des spécifications « royalty-free »**
 - Politique spécifique de propriété intellectuelle
 - Suite un processus de standardisation spécifique
 - WD, FPWD, LCWD, PR, R (fully public)



page 22




The W3C recommendations



The diagram illustrates the W3C recommendations stack, organized from top to bottom:

- Applications** (Green bar)
- Interaction** (Blue box): XHTML, CSS, SVG, SMIL, CDF, XForms, MathML, InkML
- Mobile Web** (Blue box): XHTML Basic, Mobile SVG, SMIL Mobile, XForms Basic, CC/PP, DD
- Voice** (Blue box): VoiceXML, SRGS, SSML, CCXML, EMMA
- Web Services** (Blue box): SOAP, MTOM, WSDL, WS-CDL, Addressing
- Semantic Web** (Blue box): OWL, SKOS
- Privacy, Security** (Blue box): P3P, APPEL, XML Sig, XML Enc, XKMS
- Web Accessibility / Internationalization / Device Independence / Quality Assurance** (Orange bar)
- XML, Namespaces, Schemas, XQuery/XPath, XSLT, DOM, XML Base, XPointer, RDF/XML, SPARQL** (Blue bar)
- XML Infoset, RDF Graph** (Blue bar)
- Web Architectural Principles** (Blue bar)
- URI/IRI, HTTP** (Blue bar)
- The Web Advancing to its Full Potential** (Blue bar)
- Internet** (Blue bar)

page 23





Le groupe SVG

- **Group of the W3C**
 - Part of the interaction domain
- **Buts**
 - Produire et maintenir la spécification SVG
- **Chairs**
 - Erik Dahlström (Opera Software)
 - Cameron McCormack (Mozilla)
- **W3C Team members**
 - Chris Lilley
 - Doug Schepers
- **Membres**
 - Officiellement 43 participants, réellement 8-10 participants actifs
 - Compagnies:
 - Mozilla, Opera, Google, Adobe, Microsoft, Apple, Canon, Motorola, IBM, Telecom ParisTech ...
- **Links**
 - <http://www.w3.org/Graphics/SVG/>
 - http://www.w3.org/Graphics/SVG/WG/wiki/Main_Page



page 24



SVG Roadmap of Deliverables

Roadmap of Deliverables					
Specification	FPWD	LC	CR	PR	Rec
SVG 2.0	November 2012	November 2013	March 2014	July 2014	August 2014
SVG Parameters	April 2009	December 2012	March 2013	June 2013	August 2014
SVG Integration	August 2012	December 2012	March 2013	June 2013	August 2014
Color Management	October 2009	November 2013	March 2014	July 2014	August 2014
CSS Animations	March 2009	November 2013	March 2014	July 2014	August 2014
CSS Transforms	March 2009	November 2013	March 2014	July 2014	August 2014
Filter Effects	May 2007	November 2013	March 2014	July 2014	August 2014
Masking and Clipping	January 2012	November 2013	March 2014	July 2014	August 2014
Compositing	August 2012	February 2013	April 2013	June 2013	July 2013
CSS Transitions	December 2009	November 2013	March 2014	July 2014	August 2014
Tiling and Layering	November 2012	November 2013	March 2014	July 2014	August 2014
Vector Effects	November 2012	November 2013	March 2014	July 2014	August 2014
SVG Accessibility	November 2012	November 2013	March 2014	July 2014	August 2014
Web Animations	November 2012	November 2013	March 2014	July 2014	August 2014

page 25



SVG historique

The timeline shows the evolution of SVG and related technologies. The top row represents the main SVG standard, with versions 1.0 (09/2001), 1.1 (04/2002), 1.1 2nd Edition (08/2011), and 2.0 (06/2013). The bottom row represents SVG Tiny / Mobile, with versions 1.1 (04/2002), 1.1 (05/2001), and 1.2 (12/2008).

- **Ecosystème initial:**
 - HTML 4.01: 1999
 - CSS 1.0 (2nd ed.): 1996
 - XML 1.0 (2nd ed.): 1998
- **Technologies en concurrence:** VML (Microsoft) and PGML (Adobe), Flash
- **Spécifications courante (?)**
 - SVG 1.1 (recommandation): <http://www.w3.org/TR/SVG>
- **Nouvel écosystème: intégration approfondie avec**
 - HTML 5
 - CSS 3
 - SVG 2.0 (draft): <http://www.w3.org/TR/SVG2> (<https://svgwg.org/svg2-draft/> editor's draft)

page 26

TELECOM ParisTech

SVG, qu'est-ce que c'est?

- **Vocabulaire XML pour**
 - 2D: Graphisme vectoriel
 - Avec du texte & des polices
 - Avec des primitives de dessin et de mise en page
 - Support de:
 - Style (CSS),
 - Animations (SMIL ou JavaScript),
 - Interactivité (DOM Events),
 - Script (DOM, JavaScript)
 - Images (PNG, JPG)
 - Multimedia (audio, video)
- **Libre de droit**

page 27

TELECOM ParisTech



SVG vs. Flash

■ Textuel vs. Binaire

- Plus facile à écrire 'à la main' ou par programme
- Fichiers plus gros si non compressés
- Plus facile à indexer, parcourir

■ Natif vs. Plugin

- Plus facile d'y appliquer un style
- Plus facile d'y appliquer du script
- Meilleure intégration avec HTML

page 28



XML: atouts/ inconvénients pour SVG

■ Les documents SVG peuvent bénéficier des outils du XML

- Vérification de syntaxe, validation
- Modification avec le DOM
 - More specific manipulations possible using SVG DOM or MicroDOM
- Transformations avec XSLT
- ...

■ Verbeux

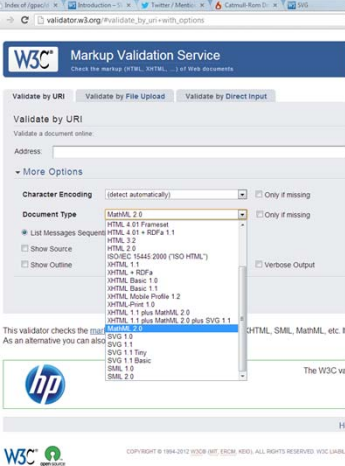
- Certains attributs sont conçus pour tenir compte de ça
- Compression

page 29



SVG Validation

<http://validator.w3.org/>



- **SVG 1.1**
 - Utilise une DTD
- **SVG 2.0**
 - XSD: surement pas
 - RNG: peut-être?
- **Les navigateurs sont robustes aux erreurs**
 - Comportement normalisé aux erreurs

page 30

TELECOM ParisTech

SVG: structure de document

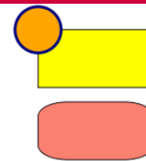
- **Racine <svg>**
 - Définit le namespace <http://www.w3.org/2000/svg>
- **SVG Document tree**
 - Dictionary of elements using <defs>
 - Grouping of elements using <g>
- **References**
 - Across the tree => graph
 - Across documents (with restrictions: image vs. script)

page 31

TELECOM ParisTech



Exemple



```
<svg xmlns="http://www.w3.org/2000/svg"
      viewBox = "0 0 500 600" version = "1.1">

  <rect x = "100" y = "100" width = "400" height =
    "200" fill = "yellow" stroke = "black" stroke-
    width = "3"/>

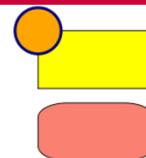
  <rect x = "100" y = "350" rx = "100" ry = "50"
    width = "400" height = "200" fill = "salmon"
    stroke = "black" stroke-width = "3"/>

  <circle cx = "100" cy = "100" r = "80" fill =
    "orange" stroke = "navy" stroke-width = "10"/>
</svg>
```

page 32



Exemple avec un groupe



```
<svg xmlns="http://www.w3.org/2000/svg"
      viewBox = "0 0 500 600" version = "1.1">

  <rect x = "100" y = "100" width = "400" height =
    "200" fill = "yellow" stroke = "black" stroke-
    width = "3"/>

  <g>
    <rect x = "100" y = "350" rx = "100" ry = "50"
      width = "400" height = "200" fill = "salmon"
      stroke = "black" stroke-width = "3"/>
    <circle cx = "100" cy = "100" r = "80" fill =
      "orange" stroke = "navy" stroke-width = "10"/>
  </g>
</svg>
```

page 33






Elément *use*

■ Réutilisation

```
<svg viewBox = "0 0 1000 1000" version = "1.1">
  <defs>
    <!-- A circle of radius 200 -->
    <circle id = "s1" cx = "200" cy = "200" r = "200" fill = "yellow"
                                   stroke = "black" stroke-width = "3"/>

    <!-- An ellipse (rx=200,ry=150) -->
    <ellipse id = "s2" cx = "200" cy = "150" rx = "200" ry = "150"
                                   fill = "salmon" stroke = "black" stroke-width = "3"/>
  </defs>
  <use x = "100" y = "100" xlink:href = "#s1"/>
  <use x = "100" y = "650" xlink:href = "#s2"/>
</svg>
```

page 34



SVG Files

■ Internet Media Type (MIME type)


- image/svg+xml

■ Extensions:

- .svg
- .svgz when compressed using GZIP
- Possible compression using EXI

page 35


Document Object Model



- **Interfaces avec l'arbre représentant le document**
 - Donne accès aux éléments d'une page web ou d'un document SVG pour lecture ou modification
 - Indépendant du langage de programmation
- **Spécifications**
 - Level 1 (spécification unique)
 - Level 2 (6 specs): Core, Style, (Views), ...
 - Level 3 (3 specs): Core, ...
- **Nombreuses APIs associées**
 - frameworks Javascript en donnent une abstraction commune
 - JQuery, Mootools, Scriptaculous, ...

page 36

TELECOM ParisTech

SVG: interfaces DOM

- **Peut utiliser le DOM générique**
- **Des interfaces additionnelles pour SVG**
 - Par exemple pour accéder à l'état courant d'une valeur animée...
- **MicroDOM**
 - version simplifiée du DOM pour SVG Tiny 1.2
 - Moins d'accès par chaînes, plus d'accès typés
 - Navigation dans les éléments
- **SVG 2.0**
 - Vers un meilleur DOM compatible avec HTML
 - Inspiré de bibliothèques JavaScript

page 37

TELECOM ParisTech



getElementById

```
var elmt = document.getElementById("monid")
```

Permet de récupérer dans une variable javascript un 'accès' à l'élément désigné par l'id fourni

Qui peut être utilisé ainsi:

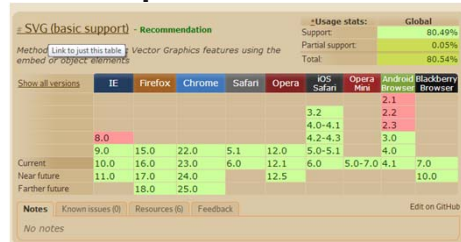
```
element.setAttribute("x", "10");
element.setAttribute("y", "20");
element.setAttribute("width", "100%");
element.setAttribute("fill", "blue");
var xelmt = element.getAttribute("x");
```

page 38



SVG: Support par les navigateurs

desktop & mobile



Source: <http://caniuse.com> (nov. 2012)

Feature	Safari on iOS	Android Browser	Google Chrome	Amazon Silk	BlackBerry Browser	Nokia Browser	Internet Explorer	Opera Mobile	Opera mini	Firefox	webOS Browser
Platform	iPhone, iPad	Phones & Tablets	Android 4.0+	Kindle Fire	Phones	Tablet	Windows Phone 7 & 8	Android & Symbian	Java/JOS Android	Android, Mac OS	HP Phones, HP TouchPad
Web Workers	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Web Workers	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Canvas API	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Canvas API	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SVG	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SVG Working Group	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Basic Vector Graphics	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Source: <http://mobilehtml5.org> (nov. 2012)

page 39





Support (suite)

- **Très bon support par les navigateurs**
- **Interopérabilité**
 - Pas de support pour les animations et SVG fonts dans IE
 - Pas de support pour SVG fonts dans Firefox
- **SVG moteurs d'affichage autres**
 - Batik SVG Toolkit (<http://xmlgraphics.apache.org/batik/>)
 - Anti Grain Geometry (<http://www.antigrain.com/>)
 - PageFire (<http://www.pagefire.com/>)
 - GPAC (<http://gpac.sf.net>)
 - Ikivo ...

page 40



Outils de création

- **Editeurs de texte**
- **Outils commerciaux**
 - Adobe: Illustrator, EdgeCode, Edge Animate, Dreamweaver
 - Microsoft: Visio
 - CorelDraw
- **Open Source Software**
 - Inkscape (GUI editor)
 - <http://inkscape.org>
 - SVG Edit (editor in the browser)
 - <http://svg-edit.googlecode.com>
- **Flash to SVG/HTML5 Converters**
 - Google Swiffy (<https://www.google.com/doubleclick/studio/swiffy/>)
 - Mozilla Shumway (<https://github.com/mozilla/shumway#readme>)
- **SVG Cleaners**
 - Inkscape: sauver comme « Plain » ou « Optimized » SVG
 - Scripts de nettoyage: scour, ...



page 41





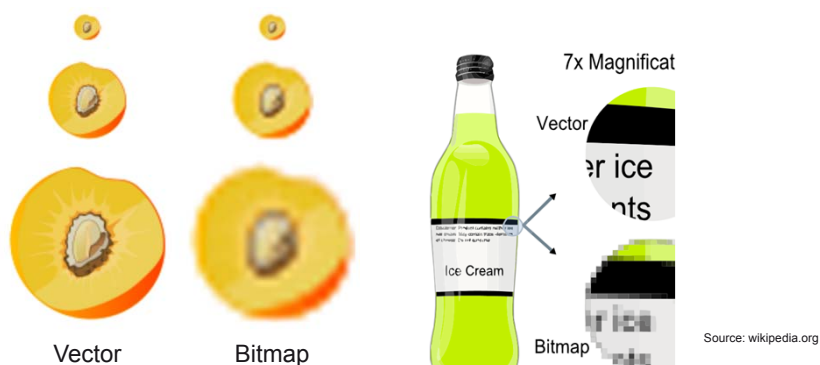
Vector Graphics?

- **Graphismes décrits par des primitives mathématiques/géométriques**
 - Objets 2D: Lines, curves, circles, rectangles, text
 - ou équivalents 3D : meshes, nurbs, spheres, ...
 - Opposition entre raster/bitmap images (description par des pixels)
- **Rendu par des pixels à une résolution donnée**
 - Zoomable proprement
 - Travail de rendu fait par le client
 - Compromis entre coût et qualité
- **Formats: SVG, VML, AI, PS, PDF, Flash...**

page 42






Bitmaps versus graphisme vectoriel



page 43



Vector Graphics & Scalability

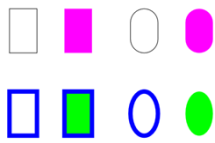
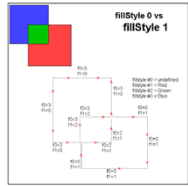
PNG				Total
	25x37 1,55 Ko	50x75 3,89 Ko	100x150 9,89 Ko	15,33 Ko
SVG	Toutes résolutions			5,93 Ko
SVGZ	Toutes résolutions			1,54 Ko

page 44

TELECOM ParisTech

Types of Vector Graphics Primitives

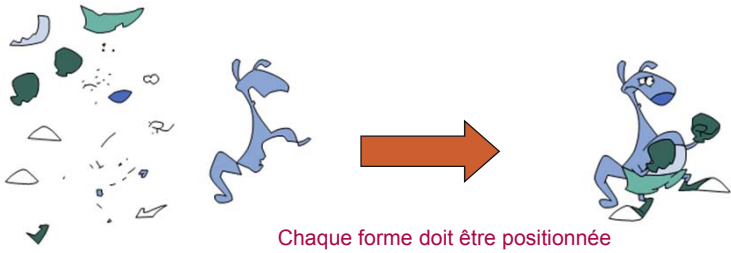
- **Contour-based representation**
 - A shape is defined by a contour
 - Can be filled with only one "paint"
 - Can be stroked with only one "paint"
- **Other representations**
 - Planar-maps (Flash SWF format)
 - A shape is defined as a list of curves with 3 "paints"
 - on the right, on the left, on the curve
 - Advanced gradients (meshes, diffusion curves)

page 45

TELECOM ParisTech

Contour-based representation of vector graphics, transformations and grouping



Chaque forme doit être positionnée

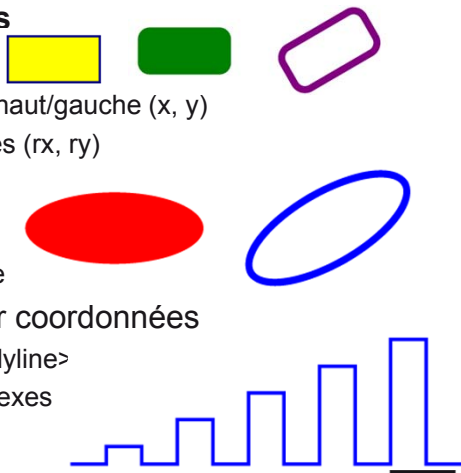
page 46

TELECOM ParisTech

Formes de base

■ Primitives graphiques

- `<rect>`
 - Accroché sur son coin haut/gauche (x, y)
 - Coins arrondis possibles (rx, ry)
- `<circle>`
- `<ellipse>`
 - Accroché en son centre
- primitives basées sur coordonnées
 - `<line>`, `<polygon>`, `<polyline>`
 - `<path>`: courbes complexes



page 47

TELECOM ParisTech

SVG Path

■ Element utilisé pour décrire des tracés complexes, libres

- <path>
- Suite de commandes de dessin composant la valeur de l'attribut « d »
 - Liste de points 2D séparés par des commandes de dessin
 - Cordonnées relatives ou absolues

■ Types de commandes

- MoveTo
- LineTo
- Close
- « Curves »
- « Arcs »
- D'autres en cours de définition pour SVG 2.0 (rotation, ...)



page 48



SVG: les courbes

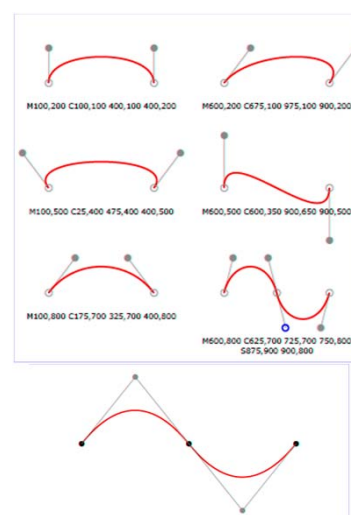
■ Courbes de Bézier

- Cubic (C)
- Cubic Symetrical (S)
- Quadratic (Q)
- Quadratic Symetrical (T)

■ Catmull-Rom Curves (SVG 2.0)

- [Dotty.svg](#) (Doug Schepers)

■ ...



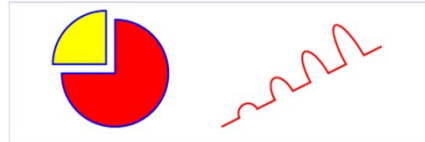
page 49





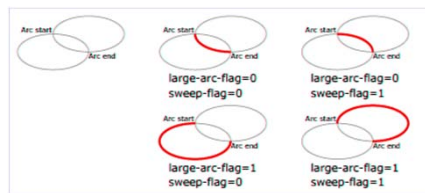
SVG: les arcs

- **Start-point, end-point + arc parameter**



- **Extensions en SVG 2.0**

- Harmonisation avec Canvas



page 50



SVG: le texte

- **SVG uses specific elements for text**

- Different from HTML
 - No flowing text
 - No paragraph
- Graphical primitives as others
 - Can be filled, stroked, ...
- With additional CSS text properties
 - @font-size, ...


TEXT IN SVG

- **SVG Text elements**

- `<text>`
 - Renders characters on a single line
- `<tspan>`
 - Used to change the style of some characters on a line
- `<tref>`
 - Reuse existing text content across `<text>` elements
- `<textPath>`
 - Draws a text along a path (ex: legend on a river)
- SVG Tiny 1.2 `<textArea>`
 - Paragraph

page 51






SVG Fonts

- **SVG allows defining its own fonts**
 - ``, `<glyph>`, ...
 - `@font-face-uri`, ...
- **SVG in OpenType**

page 52



XML



page 53

Qu'est-ce que XML?

- Une syntaxe unique...
- ... pour tout représenter!

page 54



Exemple

```

■ <book>
  • <chapter>
    - <title>Introduction</title>
  • </chapter>
  • <chapter>
    - <title>Récit</title>
    - <subChapter>
      - <title>Partie 1</title>
    - </subChapter>
    - <subChapter>
      - <title>Partie 2</title>
    - </subChapter>
  • </chapter>
  • <chapter>
    - <title>Index</title>
  • </chapter>
■ </book>

```

page 55



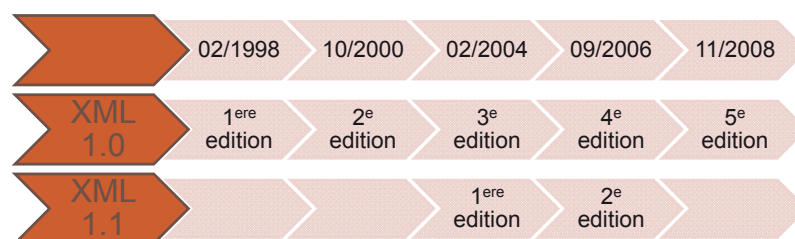


XML: eXtensible Markup Language

- XML est un système générique de représentation semi-structurée de données
- XML propose une syntaxe pour cette structuration
- XML est basé sur un principe de balisage des données
- XML a été défini par le W3C pour disposer d'un modèle de partage de données sur le Web
- XML propose un cadre commun à tous les types de documents
- XML se décline en grammaires spécialisées
 - XHTML, MathML, ...
- XML autorise les documents composés à l'aide de plusieurs grammaires
- XML et son environnement de programmation et d'outils est adapté aux transformations entre grammaires XML ou vers d'autres formats



XML



■ Recommendation W3C depuis 1998

- Langage de description d'information semi-structurée (meta-langage)
 - Exploitable par les machines
 - Lisible (relativement) par les humains



XML Syntaxe and Vocabulaire

Vocabulaire	Syntaxe
Tag d'ouverture	<monElement>
Tag fermant	</monElement>
Element 'vide'	<monElement/>
Element	<monElement/> ou <monElement>...</monElement>
Contenu textuel	<monElement>voici du texte</monElement>
Commentaires	<!-- Mon commentaire -->
Instructions	<?UneInstruction?>
Attribut	<monElement monAttribute=« valeur de l'attribut » />

page 58



Document XML

- Un document XML est un arbre
- Les nœuds de l'arbre sont des ELEMENTS qui peuvent être qualifiés par des attributs



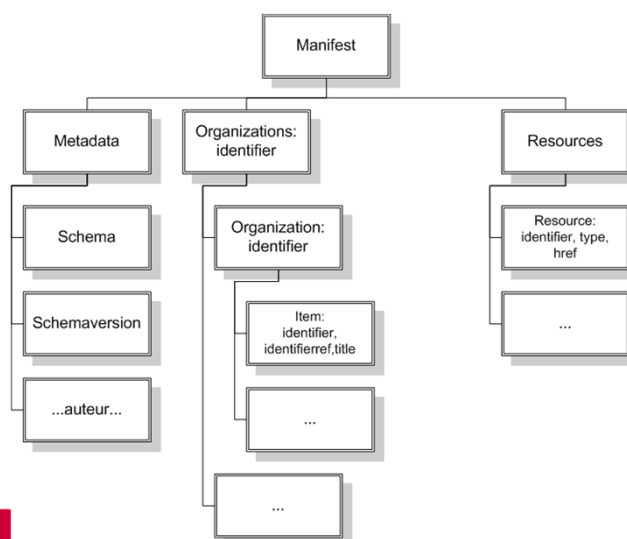


Exemple de document XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<manifest>
  <metadata> <schema>IMS Content</schema>
    <schemaversion>1.1</schemaversion>
    ... </metadata>
  <organizations>
    <organization identifier="SESSION">
      <item identifier="SLIDE1_1" identifierref="REF_SLIDE1_1"
        title="Cryp..."><datetime>0</datetime></item>
      <item identifier="SLIDE2_1" identifierref="REF_SLIDE2_1" title="Plan de l'exposé">
    ...
    </organization>
  </organizations>
  <resources>
    <resource identifier="REF_SLIDE1_1" type="webcontent" href="jpg\Slide1_1.htm">
      <file href="jpg\Slide1_1.htm" />
      <file href="jpg\Slide1_1.jpg" />
    </resource>
    <resource identifier="REF_SLIDE2_1" type="webcontent" href="jpg\Slide2_1.htm">
  ...
  </resource>
</resources>
</manifest>
```



Exemple d'arbre XML





Document XML

■ Éléments de syntaxe

- Déclaration de jeu de caractères et de type d'encodage (souvent Unicode et UTF-8)
- Langage à base de tags, d'attributs et de texte
- Contraintes sur:
 - Structure correcte
 - tags imbriqués uniquement de façon arborescente
 - `<a>` est interdit!
 - Validité (éventuellement)

■ Exemples de langages basés sur XML

- XHTML, SVG, RSS, KML...

page 62



Atouts du XML (1)

■ Syntaxe simple

- Facile à apprendre, à générer, à lire

■ Syntaxe générique

- Apprendre une fois
- Applicable à de multiples domaines/langages
- Limité en terme de sémantique

■ Syntaxe extensible

- Ajout de nouveaux éléments, attributs, valeurs
- Facilité de faire évoluer un langage basé sur XML avec une compatibilité amont et aval

■ Très large adoption

- Support unifié dans la plupart des langages de programmation
- Nombreux outils pour créer, lire, analyser, valider ...

page 63





Atouts du XML (2)

- **Une mise en œuvre légère possible**
- **Une grande souplesse d'adaptation et d'évolution**
 - Facilite la compatibilité arrière et avant
- **Naturellement multi-plateforme**
 - Format texte
- **Informations sémantiques sur le document et possibilités de recherche**
 - Le document XML est généralement composé de données encadrées par des balises qui décrivent ces données
- **Relative simplicité de passage d'une grammaire à une autre**
 - Passer des données à une présentation des données (ex: HTML)
 - Passer d'une version de fichier à une autre



Représentation semi-structurée de données

```

<dvds>
  <dvd vu="oui">
    <titre>Jules et Jim</titre>
    <realisateur>F.Truffaut</realisateur>
  </dvd>
  <dvd vu="oui">
    <titre>Plein Soleil</titre>
    <acteur>Alain Delon</acteur>
  </dvd>
  <dvd vu="oui">
    <titre>La vie de Brian</titre>
    <realisateur>Terry Gilliam</realisateur>
  </dvd>
  ...
</dvds>

```



XML: inconvénients

■ Verbeux

- Longueur des tags et des attributs
- Redondance entre les ouvertures et fermetures de tags
- Fichiers XML deviennent vite gros
 - Besoin de compression (zip, gzip, EXI)

■ Performance

- XML fichiers textes à analyser
- Le traitement de texte est lent
 - Relativement à une représentation binaire

■ Validation

- Seul un document complet est valide

page 66



XML: espaces de nommage (namespaces)

■ Buts

- Associer clairement des éléments et des attributs avec un vocabulaire:
 - Pour l'identification
 - Permet d'utiliser dans un même document des éléments et attributs de différents vocabulaires XML
 - Pour la validation

■ Principes

- Définition d'un préfixe et association avec un unique identifiant
 - xmlns:prefix = « unique identifier »
- Utilisation d'un préfixe devant les éléments et attributs
 - 'Qualified name' = prefix + ':' + local name (ex: xlink:href)

page 67





XML Namespaces examples

```
<html:html xmlns:html='http://www.w3.org/1999/xhtml'>
  <html:head>
    <html:title>Hello</html:title>
  </html:head>
  <html:body><html:p>Hello World!</html:p></html:body>
</html:html>

<bk:book xmlns:bk='urn:loc.gov:books'
          xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <bk:title>Cheaper by the Dozen</bk:title>
  <isbn:number>1568491379</isbn:number>
</bk:book>
```



ID

- Tous les éléments de tous langage XML peuvent avoir un ID
- Les IDs présents dans un document doivent être uniques
- L'ID permet de désigner un élément de façon simple et non ambiguë
- getElementbyID est une fonction de l'API standard DOM pour accéder à un élément par son ID



Grammaires XML pour le multimédia


- **XHTML**
- **SVG -> Scalable Vector Graphics**
- **SMIL**
 - Synchronized Multimedia Integration Language
- **HTML+TIME -> HTML+déroulement temporel**
- **XMT -> Représentation XML de MPEG-4 BIFS**
- **InkML -> annotations de documents**
- **MPEG-7, MPEG-21**



Grammaires associées



- Entre autres:**
- **IMS/Scorm**
 - Représentation de ressources pédagogiques
 - **Trans-13**
 - Représentation de transcriptions
 - **MathML**
 - Représentation de formules mathématiques
 - **MusicML**
 - Représentation de partitions
 - **XUL, XAML...**
 - Représentations d'interfaces utilisateur
 -





Grammaires utiles

- **SOAP**
 - Communication de messages entre applications
- **RDF**
 - Méta-données
- **XSLT, XSL-FO**
 - Transformations de documents XML




CSS



page 73

CSS: Cascading Style Sheets



- **Langage servant à associer un style de présentation à des documents**
 - But: séparer le 'contenu' de la présentation
 - Spécification complémentaire de HTML et SVG
 - mais peut être appliquée à n'importe quel type de document structuré sous forme d'arbre (e.g.XML)
- **Deux spécifications**
 - CSS 2.1: corrections et dépréciation de CSS 2
 - CSS 3: spécification modulaire de CSS 2.1 & nombreuses additions (50+ modules en 2012)

page 74

TELECOM ParisTech

CSS: Principes

- **Langage basé sur des règles associées à des éléments d'un document**
 - propriétés visuelles (fond, bordure, ...)
 - propriétés spatiales (mise en page, ...)
 - propriétés des textes (polices, ...)
- **Ensemble de règles est un 'style sheet' (feuille de style)**
 - Plusieurs styles peuvent être définis pour un documents
 - style sheet utilisateur
 - style sheet auteur
 - Style sheet pour un type de device
 - Méthodes de sélection de style sheet (CSS Media Queries)

```
<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css">
<link rel="stylesheet" type="text/css" media="print" href="serif.css">
```

page 75

TELECOM ParisTech



CSS: Syntaxe

■ Rules = selectors + declaration block (use of { })

■ Selectors

- Addressage d'éléments dans un arbre
- Ex: *, element[foo], element[foo='bar'], element.class, ...
- Pseudo-classes (e.g. :hover), pseudo-elements (e.g. :first-line)

■ Declaration = propriété + ':' + valeur

■ Héritage de valeur de propriété

```
p {
  border-style:solid;
  border-width:5px;
}
```

```
a:link {color:#FF0000;}
a:visited {color:#00FF00;}
a:hover {color:#FF00FF;}
a:active {color:#0000FF;}

```

page 76



HTML



page 77

HTML « HyperText Markup Language »

The diagram shows the evolution of HTML and XHTML. The top row represents HTML versions: 1.0 (1993), 2.0 (11/1995), 3.2 (01/1997), 4.0 (12/1997), 4.01 (12/1999), 5 (2014 ?), and 5.1. The bottom row represents XHTML versions: 1.0 (01/2000), 1.1 (05/2001), 1.0 2nd Edition (08/2002), and 2.0 (X). Arrows indicate the progression of time and the relationship between the two standards.

- **Langage à base de balises des pages web**
 - version initiale créée par Tim Berners-Lee en 1989
 - Plusieurs variantes
 - HTML 4 Strict, Transitional, Frameset
 - HTML vs. XHTML
 - HTML 5 (HTML.next, HTML 5.1)

page 78

TELECOM ParisTech

HTML5

- **Version en construction de la recommandation**
 - Initialement développée par le groupe WHATWG
 - Désormais suit le canal des recommandation W3C
- **Specification massive et modulaire**
- **Avec des nouveautés importantes**
 - Association de SVG et MathML dans des pages HTML
 - Multimedia: éléments <video>, <audio>, <track>
 - Graphics: élément <canvas> & API associée
 - Offline Storage Database, Web Storage
 - Drag-n-drop
 - ...

page 79

TELECOM ParisTech



W3C SVG Scalable Vector Graphics

Détails et extensions à venir



page 80

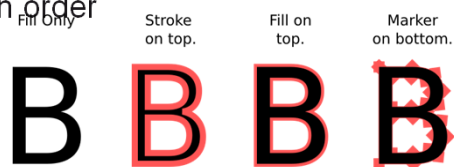


SVG Rendering Model

■ Individual element rendering

- Drawing operations in order

- Fill
- Then stroke
- Then markers
- Then filters
- Then clip
- Then mask



- Order to be parametrized in SVG 2.0

page 81





Grouped elements rendering

■ Algorithm du peintre:

- Les éléments sont dessinés dans l'ordre du document en utilisant un traitement simple des superpositions
- Extensions à venir

■ Possibility to use z-index to change element order in SVG 2.0

page 82



SVG Painting Properties: Fill Properties

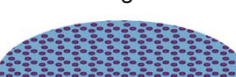
■ @fill

- A uniform/solid color
 - sRGB color space or ICC color profile
 - Extensions in SVG 2 Color Module
 - Syntax:
 - `rgb(int[0-255], int[0-255], int[0-255])`
 - `rgb([0-100]%, [0-100]%, [0-100]%)`
 - Color keywords: black, white ...
- A linear or radial gradient
 - Also used in CSS
 - Extensions to Gradient Meshes in SVG 2
- A pattern
 - Extensions to hatches in SVG 2
- **@fill-opacity**
 - Transparency used for alpha-blending
- **@fill-rule**
 - When a graphical primitive self-intersects

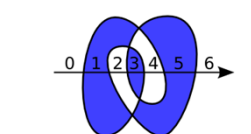
Linear gradient



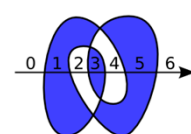
Radial gradient



Pattern



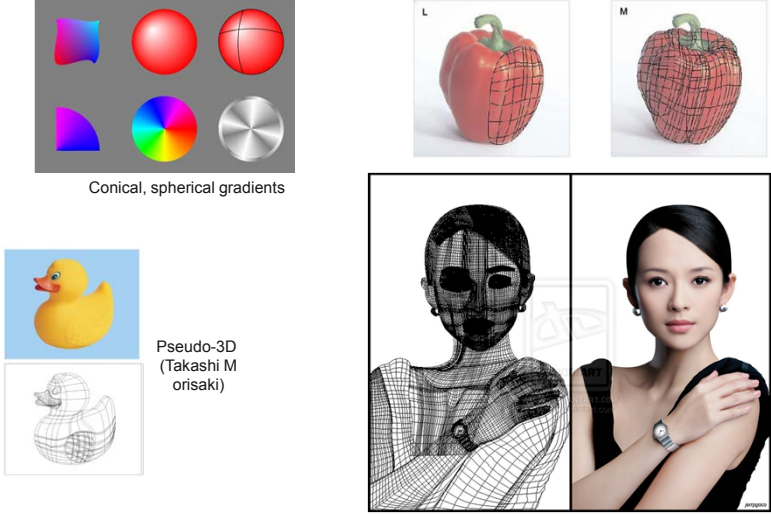
Fill rule



page 83



Gradient Mesh (exemples)



Conical, spherical gradients

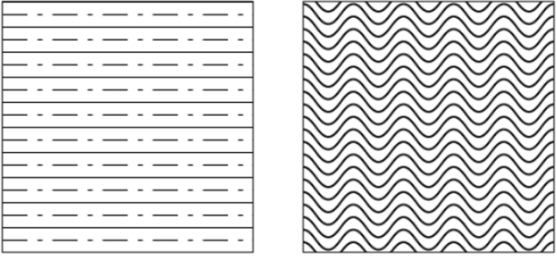
Pseudo-3D
(Takashi Morisaki)

TELECOM
ParisTech

page 84

Hatches

■ SVG 2.0 feature to come

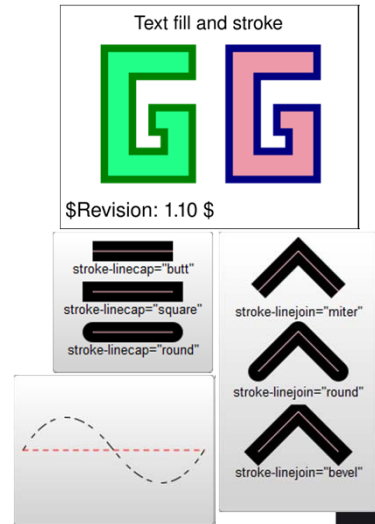


TELECOM
ParisTech

page 85

SVG Painting Properties: Stroke Properties

- **@stroke**
 - Same syntax/values as @fill including gradients, pattern, ...
- **@stroke-opacity**
 - Same as fill-opacity but only on the stroke
 - Can be combined
- **@stroke-width**
 - Centered around the mathematical/geometrical outline
 - New attribute in SVG 2.0 to control the position of the stroke
- **@stroke-dasharray**
- **@stroke-dashoffset**
- **@stroke-linejoin**
- **@stroke-linecap**

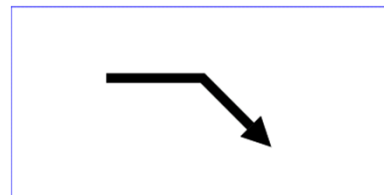
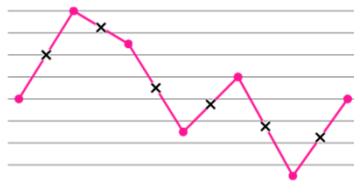


page 86



SVG Markers

- **Draws a symbol at some specific locations of a given graphical primitive**
 - Initially for point-based graphical primitives (path, line, polygon ...)
 - To be extended to all primitives (rect, circle, ...) in SVG 2.0
 - Initially at specific positions: start, end, middle
 - To be extended to any position (%)



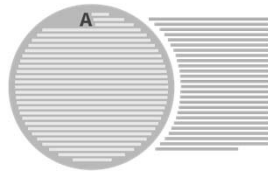
page 87





CSS Exclusions

- To exclude certain regions from the text flow
- <http://html.adobe.com/webstandards/cssexclusions/>

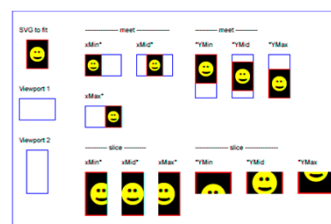


page 90



Viewing SVG graphics: Vocabulary

- **SVG canvas**
 - Infinite rendering area onto which the graphical elements are rendered
- **SVG viewbox**
 - Rectangular region clipping the canvas defined by the viewbox attribute (x y width height)
- **SVG window / viewport**
 - How the viewbox is viewed
 - fitted or sliced to a size width x height
 - Defined by the width, height and preserveAspectRatio attributes
 - Treated as if the viewbox was an image (also used for images)
- **Possible enhancement in SVG 2.0**
 - Automatic viewbox corresponding to the bounding box of the objects on the canvas



page 91





Standalone viewing vs. Embedded viewing

■ Standalone viewing

- Width/height with fixed unit: the player creates a window of that size
- Width/height relative units (e.g. 100%): the player allocates a size

■ Embedded viewing

- In HTML using CSS
- The viewport is defined by the CSS Box Model

page 92



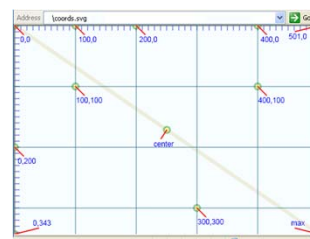
Positioning of Graphics

■ Two major types of spatial organization

- 2D or 2.5D:
 - Objects are positioned with 2 floating-point coordinates (+ possibly a integer layer information)
 - HTML+CSS, Flash, SVG, SMIL, BIFS
- 3D:
 - Objects are positioned with 3 floating-point coordinates
 - VRML, WebGL

■ Each type of spatial organization defines

- The local coordinate systems for each type of object
 - Where the (0,0,0)-point is
 - Used for positioning, rotation, scaling ...
- The nesting of local coordinate systems
 - Transformation from one to another
 - E.g. CSS Box Model, SVG Transformations



page 93




Example of Local Coordinate Systems

■ **Graphics**

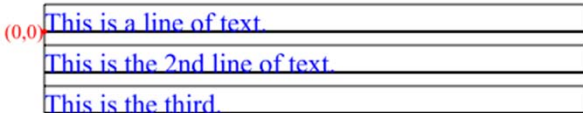
```
<path stroke="black"
  d="M 100 100 L 200 200"/>
```

■ **Text**

```
<text x="0" y="100" font-size="80"
  fill="red">Doing text</text>
```



The diagram illustrates two coordinate systems. The top part shows a path from (100, 100) to (200, 200) on a grid. The bottom part shows the text 'Doing text' in red, positioned at (0, 100) relative to the path's start point. The text's bounding box is shown with coordinates (6.15, 42.72) to (359.48, 116.79).



The diagram shows three lines of text stacked vertically. The first line is 'This is a line of text.' and the second is 'This is the 2nd line of text.' The origin (0,0) is marked at the top-left of the first line.

page 94

SVG Coordinate Systems

- **Global Coordinate System:**
 - Origin: default top-left of the viewBox
 - X-axis right-wards, Y-axis downward
- **Local Coordinate Systems**
 - Origin: typically top-left or center of the shape
- **Intermediate Coordinate Systems**
 - <g> elements
- **Units**
 - No precision limit
 - Many possible units from CSS: cm, px, em, ...
 - User unit
 - Relation to device pixels

page 95



HTML and SVG integration

■ HTML in SVG:

- Use of SVG `<foreignObject>` element

■ SVG in HTML

- Reference from an HTML document
 - Older versions of browsers:
 - `<embed>`, `<object>` or `<iframe>`
- Inclusion (mixed namespaces) in the HTML document
- Future versions ?
 - Real mix between element from different namespaces
 - `<html:p>some text <svg:circle .../><html:image ...></html:p>`

page 96




SVG in HTML

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <body>
    <div>
      <svg xmlns="http://www.w3.org/2000/svg"
          width="800px" height="600px" viewBox="0 0 800 600">
        ...
      </svg>
    </div>
  </body>
</html>
```

page 97






Usage of Media in SVG

- **Support for images in all versions**
 - JPEG
 - PNG
- **Support for specific <audio>/<video> only in SVG Tiny 1.2**
- **Use of HTML 5 <audio>/<video> in SVG 2.0**

page 98

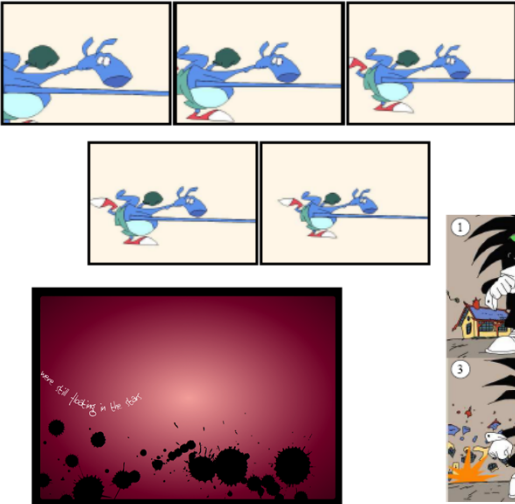


Timing et animation



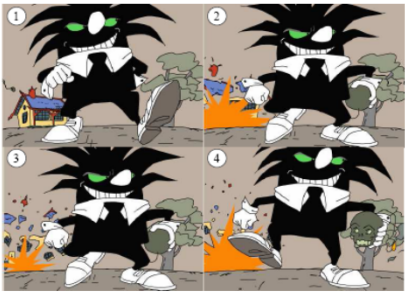

page 99

Animation



■ Applications


- Dessin animé, publicité
- Utilisé aussi pour le sous-titrage, le suivi d'objets...

page 100

Graphics Scene and Timing organization

- **Need to describe the evolutions of the scene**
 - Changes between scene states: discrete or continuous, data or structural changes ...
 - As a result
 - of animations
 - of timed server modifications
- **Two possible descriptions**
 - All scene states are provided in one description
 - Self-contained Animations: Interpolation, client-driven
 - Suitable for offline usage or heavy-client approaches
 - Non-self contained animations
 - Scene states are transmitted separately
 - Need to get additional states (push, pull) Push/Pull
 - Suitable for progressive or streamed transmission, light-client



page 101

Self-Contained Animation Description

```

<smil>
  <body>
    <seq repeatCount="indefinite">
      
      
    </seq>
  </body>
</smil>

<svg>
  <rect width="120" height="50" fill="blue" >
    <animate begin="10s" dur="2s" repeatCount="5" attributeName="fill"
      from="blue" to="red" fill="freeze" />
  </rect>
</svg>

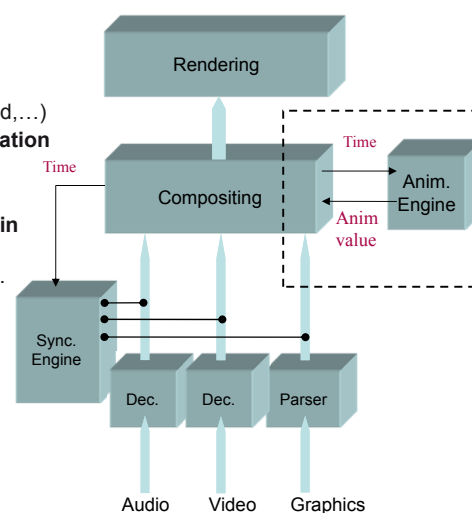
```

page 102



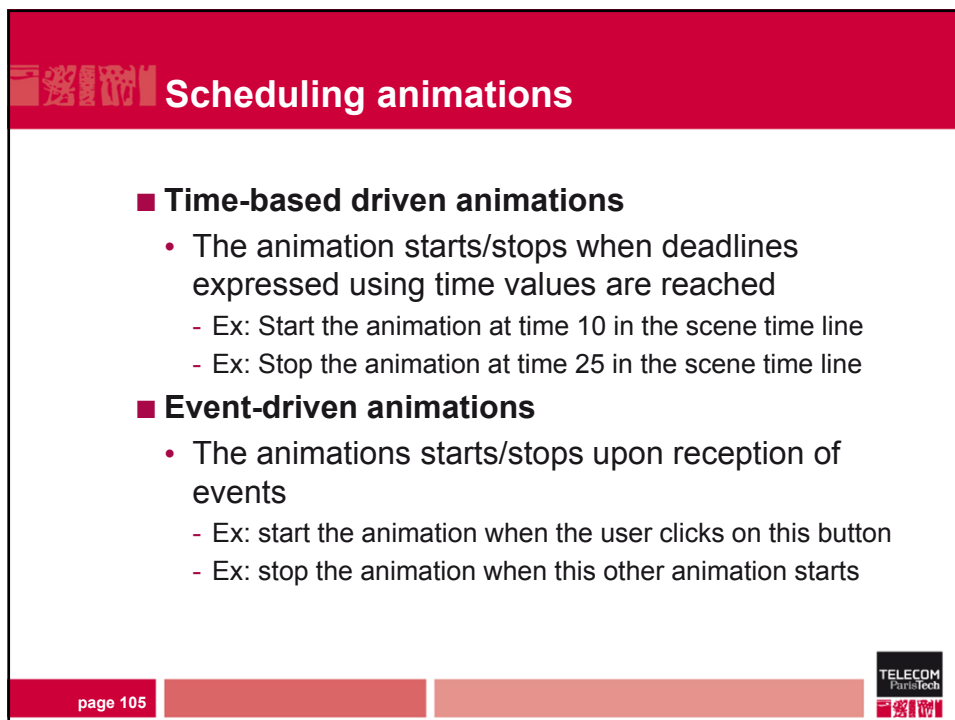
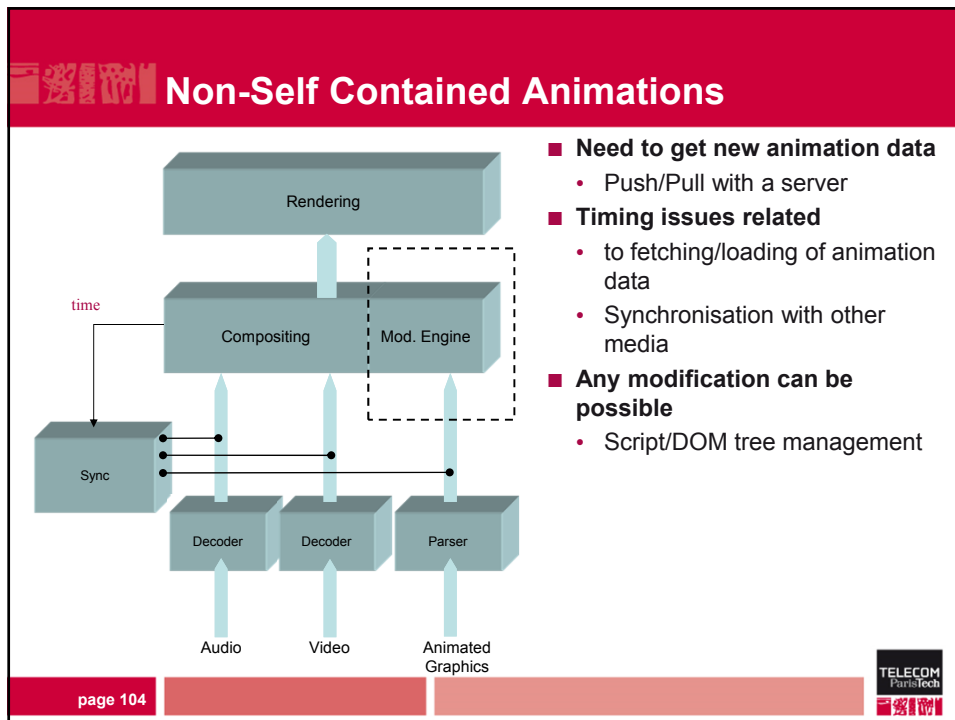
Self-Contained Animations

- **Time handled natively by the player/browser**
 - Variable refresh rate (CPU load,...)
- **Typically animation by interpolation**
 - Simple: discrete ou linear
 - Complex: spline
- **Simple modification of a value in the graphics**
 - Size, position, display on/off ...



page 103







Scripted animations

■ Use of callback functions and timers

- Ex: using the 'Window' object
- Ex: using an 'SVGTimer' object
- Ex: using requestAnimationFrame

■ Management of the synchronization by the script

```
<rect id='R' width="120" height="50" fill="blue">
<script>
function doAnimation(){
  var rect = document.getElementById('R');
  x=x+xincr;
  rect.setAttribute('x', x);
  window.setTimeout("doAnimation()", 10);
}
</script>
```

page 106



Synchronization in animations and media

■ Time lines

- Each media element has a clock and an associated time line
- The scene itself has a clock and associated time line

■ Synchronization properties describes how all clocks relate

■ Clocks can be

- Independent: No synchronization.
- Locked: if one is paused, the other is too. If the pace changes (e.g. speed), the other's pace is changed too.
 - Hard synchronization.
 - Tolerance in the synchronization

page 107





SVG Timing

- **SVG uses a restricted SMIL Timing**
 - Timed elements: animation elements and media elements
 - Timed containers
 - SVG has only one time container (one time line per document)
 - Media elements (in Tiny 1.2) define their own time lines, with simplified clock management
- **Attributes to control the timing**
 - Common to media (Tiny 1.2) and animations
 - Attributes « begin », « end », « dur », « repeatDur » ...
- **Attributes to control the synchronization (Tiny 1.2)**
 - Attributes « syncBehavior », « syncTolerance »

page 108



SVG Animations

- **SVG reuses SMIL animation as is**
 - Simple and complex animations possible
 - Definition of the interface with CSS (Sandwich animation model)
- **Elements for animations**
 - <animate>, <animateColor>, <animateMotion>, <animateTransform>, <set>
- **Attributes for animations**
 - « xlink:href », « attributeName »: to define the target attribute to animate
 - « from », « to », « values »: to give the interpolation values
 - « calcMode », « keyTimes », « keySplines »: to give the interpolatino type
 - « additive », « accumulate » : to give the behavior when multiple animations target the same attribute

page 109





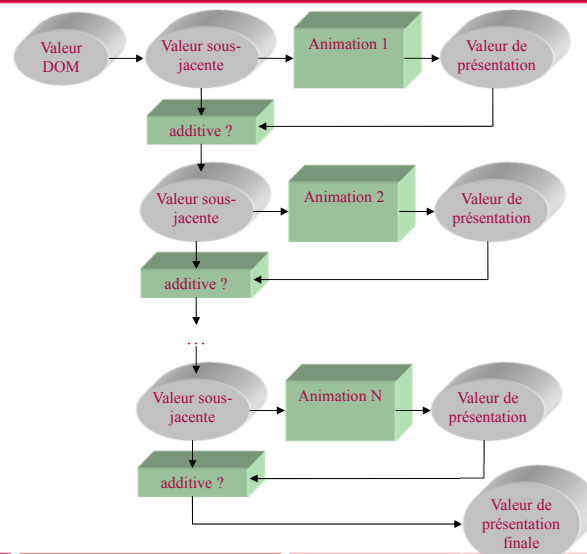
SVG animation examples

```
<g id="AnimationPaths">
  <path id="Zuerich_Geneva" stroke="orange" stroke-width="2000"
    fill="none" d="M682500,53500 C632500,53500 549500,80000
    499500,181000" />
  <use id="AirplaneZurichGeneva" xlink:href="#airplane">
    <animateMotion id="animMotionZurGen" dur="&animDuration;"
      repeatCount="indefinite" rotate="auto-reverse"> <mpath
      xlink:href="#Zuerich_Geneva"/>
    </animateMotion>
    <animateTransform attributeName="transform" type="scale"
      keyTimes="0;0.2;0.8;1" values="1.5;4;4;1.5"
      dur="&animDuration;" additive="replace" fill="freeze"
      repeatCount="indefinite"/>
  </use>
</g>
```

page 110



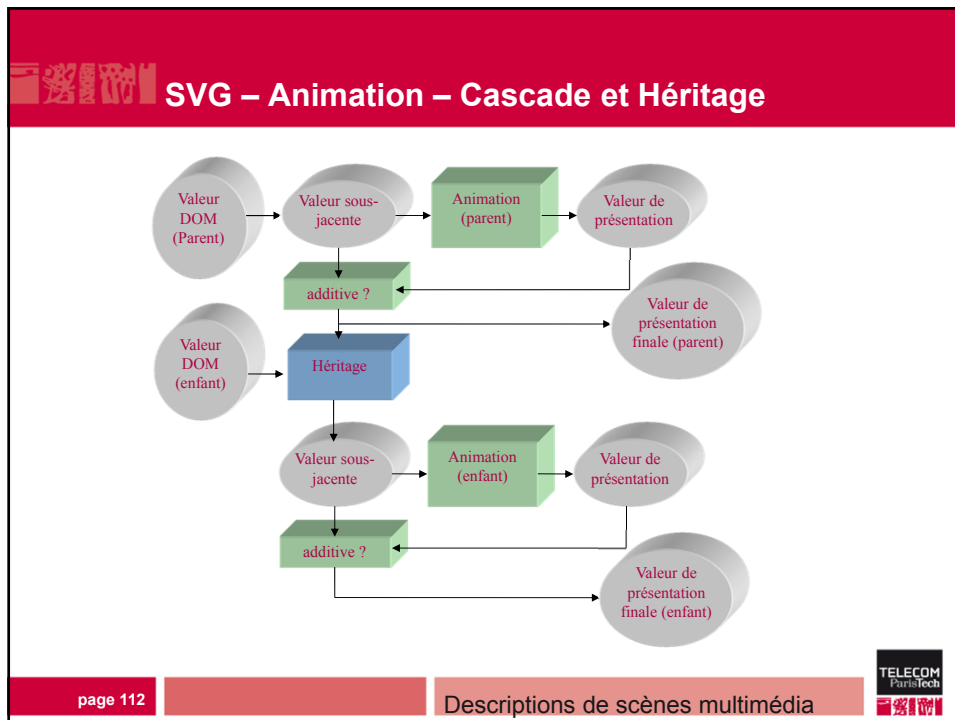
SVG – Animation – Cascade



page 111

Descriptions de scènes multimédia





CSS Animations

- **Declare animations in CSS**
 - Animation is started (offset) when the style is attached
 - Different model from SVG
- **Properties animation-***
 - name, duration, timing-function, iteration-count, direction, play-state, delay
 - Keyframe values
- <http://dev.w3.org/csswg/css3-animations/>

```

div {
  animation-name: diagonal-slide;
  animation-duration: 5s;
  animation-iteration-count: 10;
}
@keyframes diagonal-slide {
  from { left: 0; top: 0; }
  to { left: 100px; top: 100px; }
}
  
```

TELECOM ParisTech

page 113



CSS Transitions

- **CSS declaration of smooth value changes between 2 states**

- <http://dev.w3.org/csswg/css3-transitions/>

- **CSS Properties: transition-***

- property, duration, timing-function, delay

Ex: values before and after are not known

```
div {
  transition-property: opacity;
  transition-duration: 2s;
}
```

page 114



Web Animations

- **Goals**

- Provide a common foundation for CSS and SVG animation
- Provide a high quality script API for animation and synchronization
 - Easily create sophisticated frame-rate independent animations
 - Amenable to hardware acceleration
- Provide a single target for future animation work
- Advance the set of animation features available on the Web platform

- **Single Model with different syntaxes**

- JavaScript API
- CSS Syntax
- SVG Syntax

- <https://dvcs.w3.org/hg/FXTF/raw-file/tip/web-anim/index.html>

page 115





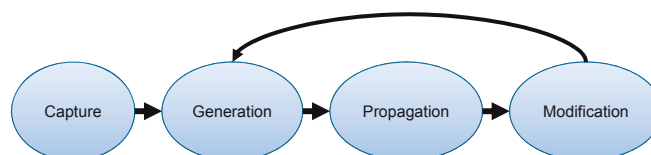
SVG : Interactivité

page 116

Interactivity Principles

■ Modèle d'évènements = 4 phases

- Capture d'un évènement utilisateur
 - Utilisation d'un 'listener'
 - Dépend du type de terminal et de la nature de l'entrée
- Génération d'un évènement dans la scène
 - Evènement simple (tel que 'isClicked')
 - Evènements complexes (objet Javascript avec propriétés)
- Propagation de l'évènement dans l'arbre de scène
 - Jusqu'à l'élément qui va gérer l'évènement (via le 'handler')
- Signaler les modifications à la scène



page 117





Interactivité : Script ou pas

■ Modifications par script

- Utilisation de ECMAScript + API spécifique
 - Web standards => JavaScript (DOM API, DOM Events, ...)
 - Flash => ActionScript
 - VRML, BIFS => VRMLScript

■ Modifications sans script

- Déclenchement d'une animation, d'une navigation

page 118



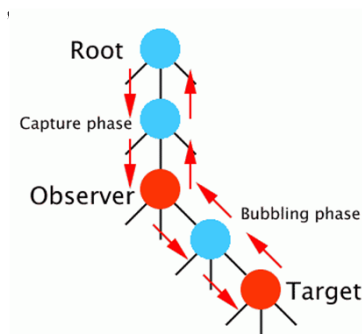
DOM Events et XML Events

■ Un modèle de propagation d'évènement

- Phase de capture, phase 'cible', phase de propagation
- Annulation d'évènement,
- Action par défaut

■ Deux spécifications W3C

- DOM Events
 - API défini le traitement des évènements en JavaScript
- XML Events
 - syntax générique pour déclarer des 'listeners', 'handlers' ... en XML
 - Compatible avec les DOM events



page 119



Exemples : code 'DOM Events'

```
<script type="application/ecmascript" >
  function doSomething(evt) {
    ...
  }
</script>
<text onclick="doSomething(evt)" >Hello World!</text>
```

```
<script type="application/ecmascript" >
  function doSomething(evt) {
    ...
  }
  e = document.getElementById('T');
  e.addEventListener('click', doSomething, false);
</script>
<text id="T" >Hello World!</text>
```

page 120

TELECOM
ParisTech

DOM Events

- **Référence**
 - https://www.w3.org/standards/techs/domevents#w3c_all
- **Progress Events (XMLHttpRequest ou Media)**
- **XML Events**
- **Pointer Events**
- **Touch Events**
- **UI Events (focus, keyboard, mouse, wheel,**

page 121

TELECOM
ParisTech

XML Events en SVG Tiny 1.2

```
<script type="application/ecmascript" >
  function doSomething(evt) {
    ...
  }
</script>
<circle cx="300" cy="225" r="100" fill="red">
  <handler type="application/ecmascript" ev:event="click">
    doSomething(evt);
  </handler>
</circle>
```

page 122



SVG : Interactivité

- **Toutes les versions de SVG supportent ECMAScript**
 - les API diffèrent
- **SVG 1.1 utilise**
 - APIs DOM : DOM Core, DOM Events, ...
 - Et, en complément, SVG DOM
 - Utilise @onclick, @onmouseover ...
- **SVG Tiny 1.2 utilise**
 - Interfaces DOM typées (nommées Micro DOM)
 - Ex: setFloatTrait au lieu de setAttribute
 - XML events: <listener>, <handler>
- **SVG 2 va simplifier cela et réutiliser le modèle interactif de HTML**

page 123





AJAX “Asynchronous JavaScript and XML”

■ Combine l’usage de différentes technologies

- HTML ou SVG
- ECMAScript
- XML
- HTTP Download

■ Exemple

- HTML/SVG + XML + DOM + XMLHttpRequest

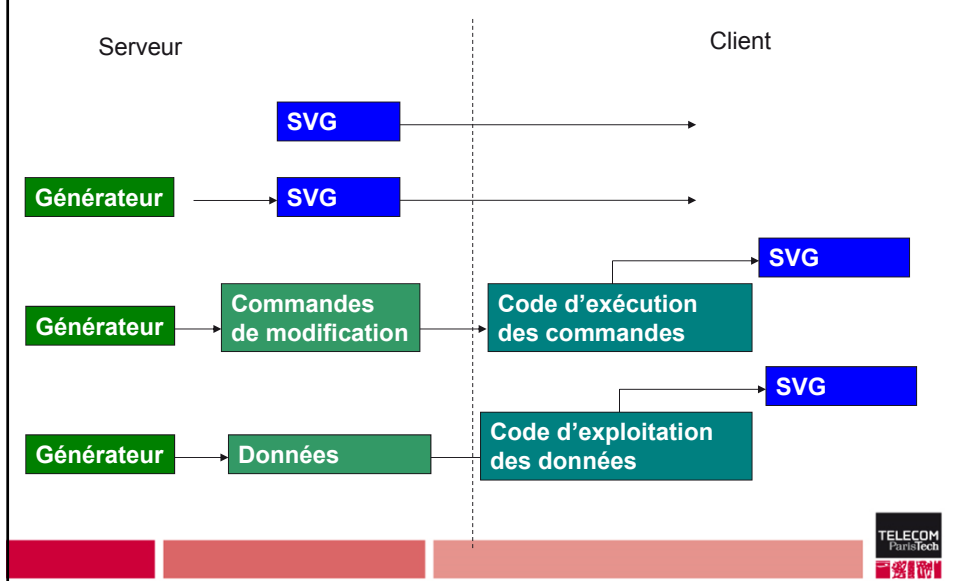
■ Bénéfices


- Les requêtes sont asynchrones/rendu
 - Evite l'attente de la réponse pour tracer ou interagir
- Permet une interactivité importante gérée côté client
 - Avec accès à des bases de données...

page 124




Utilisations de SVG





Utilisations de SVG

- **Statique**
- **Dynamique**
 - Création côté serveur en PHP, PERL, JSP...
 - AJAX avec envoi de commandes
 - Voir <http://svgmpeg4.blogspot.com/2006/07/un-zeste-de-svg-ajax-et-rex.html>
 - AJAX avec envoi de données



Processus de tracé et 'affichage



page 127

Visualisation d'Applications Multimedia

```

graph LR
    Chargement --> Composition
    Composition --> Tracé
    Tracé -.-> Chargement
  
```

- **Architecture Simplifiée**
 - Chargement : création d'une représentation mémoire
 - Composition: application des styles, traitement des scripts (scripting), animations ... pour produire une représentation prête pour l'affichage
 - Tracé: gestion de la carte graphique, des accélérateurs, de la carte videoaudio,...
- **Plusieurs tâches**
 - Synchrones (ou pas)
 - Cycliques (ou pas)

page 128

TELECOM ParisTech

Visualisation & Représentation en mémoire

- **Consommation de mémoire**
 - Dépend du nombre d'objets dans la scène
 - Du contenu SVG peut avoir des milliers d'objets graphiques (cartes, dessin animé...)
 - Dépend du choix de la structure de représentation
 - Ex: CSS Tree vs. DOM Tree
 - Dépend du compromis entre vitesse et mémoire
 - Ex: garder en mémoire des objets compressés, décompressés au vol quand on les utilise

page 129

TELECOM ParisTech



Composition

■ Principes

- Utilisation de plusieurs sources de données
 - Fichiers/flux réseau, script (AJAX, ...)
 - Différents types de media
- Besoin de 'composer' la video et l'audio
 - Prise en compte des animations, de l'interactivité
- Synchronisation entre médias

■ Facteurs de performance

- Nombre d'objets à composer
- Détection des changements d'un cycle de composition au suivant

page 130



Tracé/rendu

■ Rendu visuel

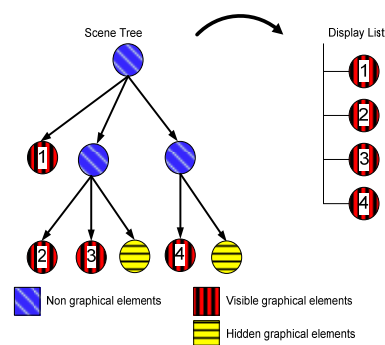
- Rasterizing 2D vector graphics & text
- Resampling
- Merging synthetic and natural images (transparency, color spaces, ...)

■ Audio Rendering

- Mixage des sons

■ Problèmes de Performance

- Antialiasing
- mode de rendu immédiat vs. différé
 - Détection des 'boîtes' à redessiner
 - Nombre de pixels à retracer à chaque cycle

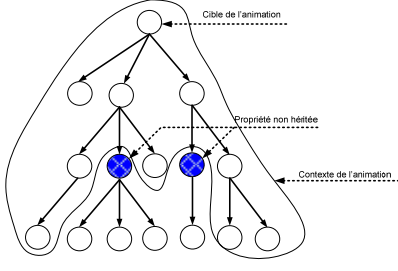


page 131



Rendu d'animations, de styles et héritage d'attributs

- Une animation peut modifier un sous-arbre
 - Besoin d'un suivi efficace des portions de l'arbre modifiée et des zones d'affichage correspondantes



The diagram shows a tree structure with nodes represented by circles. A dashed line outlines a portion of the tree, labeled 'Cible de l'animation' (Animation target). Within this target, two nodes are highlighted with blue circles and labeled 'Propriété non héritée' (Non-inherited property). The area outside the dashed line is labeled 'Contexte de l'animation' (Animation context).

TELECOM
ParisTech

page 132



Canvas



page 133

Gestion avec JavaScript de graphisme vectoriel


■ Canvas 2D

■ Canvas vs. SVG, Canvas + SVG

- Performances de tracé
- Qualité du tracé: Anti-aliasing
- Scalabilité du tracé

```
function drawPicture(){
  var canvas = document.getElementById('example');
  var context = canvas.getContext('2d');
  context.fillStyle = "rgb(0,255,0)";
  context.fillRect (25, 25, 100, 100);
  context.fillStyle = "rgba(255,0,0, 0.6)";
  context.beginPath();
  context.arc(125,100,50,0,Math.PI*2,true);
  context.fill();
  context.fillStyle = "rgba(0,0,255,0.6)";
  context.beginPath();
  context.moveTo(125,100);
  context.lineTo(175,50);
  context.lineTo(225,150);
  context.fill();
}
```

page 134



Canvas: une API de tracé


■ Objet javascript prédéfini

- Donne accès à une zone de tracé en mode pixel

■ Fonctions associées permettant les tracés

- Contexte graphique: couleur courante...
- Fonctions de tracé (cercle, trait...)

page 135





Canvas: code de base

```
<html>
<head><title></title></head>
<body>
  <canvas id="tuto" width="500" height="100"
    style="border: 1px solid;"></canvas>
  <script language="javascript">
    var canvas = document.getElementById('tuto');
    var ctx = canvas.getContext('2d');
    // Le code javascript de tracé ira ici
  </script>
</body>
</html>
```

page 136



Canvas: contexte graphique

■ Exemples:

- `ctx.fillStyle = "rgb(0,255,0)";`
- `ctx.fillStyle = "rgba(255,0,0, 0.6)";`
- `ctx.strokeStyle = ...`

page 137





Canvas: tracé 'libre'

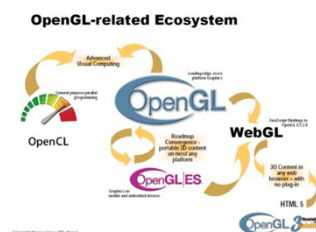
```
ctx.beginPath();           // commence un tracé
ctx.moveTo(0, 20);         // définit le 'stylo' sans tracer
ctx.lineTo(50, 30);        // prépare une ligne
ctx.stroke();              // termine le tracé
```

page 138



Rendu avec GPU

■ WebGL



■ Nvidia Path Rendering extensions



page 139





Visualisation de données



page 140

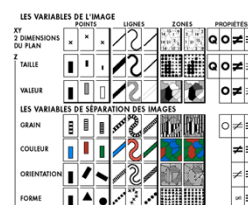
Visualisation de données

■ Un domaine riche et ancien

- La carte comme un mode de visualisation de données
- Bertin et ses successeurs
 - Expressivité des modes de présentation

■ Fort intérêt

- Pour la visualisation scientifique
- Pour le traitement des masses de données
 - Le traitement peut-être la visualisation elle-même



page 141





Outils pour la visualisation

■ Des outils 'démocratisent' la visualisation de données

- Permettent de simplifier les aspects techniques de la visualisation
- Nécessitent généralement un travail conceptuel pour la qualité visuelle de la représentation

■ Nombreux outils pour le Web

- Graphiques, cartes, animations...

page 142



JavaScript Frameworks

■ Cartes

- Google Maps API

■ SVG-based framework

- D3.js <http://d3js.org/>
- Raphael.js [http://raphaeljs.com/](http://dmitrybaranovskiy.github.io/raphael/)
- Processing.js: <http://processingjs.org/>
- SVGWeb: <https://code.google.com/p/svgweb/>

■ Canvas-based Frameworks

- Canvg <https://code.google.com/p/canvg/>
- Fabric.js <http://fabricjs.com/>

page 143



D3.js

- **Data-Driven Documents**
- **Librairie javascript**
- **Permet de déporter la construction de la présentation des données par la machine du client**
- **Simplifie pour le développeur la définition de la visualisation**
- **Importante communauté(s) d'utilisateurs**
 - Nombreux exemples, tutoriels, forums...

page 144



D3.js, catalogue d'exemples

Visual Index



page 145



D3.js, page vide modèle

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3 Page modele</title>
    <script type="text/javascript" src="../d3/d3.v3.js"></script>
  </head>
  <body>
    <script type="text/javascript">
      // votre code D3 ici
    </script>
  </body>
</html>
```

page 146



D3.js, un code avec des données CSV (1)

```
var dataset; // variable globale
function genererVis() { ... }
d3.csv("exemple.csv", function(data) {
  dataset = data;
  genererVis();
});
```


page 147



D3, visualisation avec les données: genererVis()


```
function genererVis() {  
  var svg = d3.select('body')  
    .append('svg')  
    .attr('width', 800)  
    .attr('height', 500);  
  
  var circles =svg.selectAll('circle')  
    .data(dataset)  
    .enter()  
    .append('circle');  
}
```

page 148



```
circles.attr('cx', calcPosH(d, i));  
circles.attr('cy', 200);  
circles.attr('r', calcRayon(d));
```

page 149



D3: utiliser les données pour gérer des paramètres graphiques

■ Suite de genererVis

```
circles.attr('cx', function(d, i) {
    return (i * 60) + 60;
})
.attr('cy', 200)
.attr('r', function(d) {
    return d["Quantités"];
});
```

TELECOM
ParisTech

page 150

Fait avec D3 et json

TELECOM
ParisTech

page 151



Interfaces utilisateur et graphisme vectoriel 2D animé et interactif



IHM et scène multimédia

- Une interface utilisateur peut être vue comme une scène multimédia interactive où tout ou partie des interactions sert à déclencher des actions
- Distinguer
 - Les possibilités d'interaction
 - Les possibilités de relier les interactions à des actions
 - ActionScript (Flash), tous langages dont Javascript (SVG DOM), Java (SVG JSR 226, MPEG-4 BIFS)



Tendances

■ Interfaces utilisant du graphique vectoriel 2D

- Avalon, KDE/SVG, XUL/SVG
- Sites Web en Flash
- Possibilités graphiques dans XAML

page 154



SVG et Interfaces

■ SVGGUI

- Projet Sourceforge
- <http://sourceforge.net/projects/svgui/>

■ Kevlindev GUI

- <http://www.kevlindev.com/gui/>

■ CGUI

- <http://homepage.usask.ca/~ctl271/cgui/>

page 155





Pourquoi du vectoriel dans les interfaces

■ Rappel

- Pour des raisons de performances, les éléments graphiques d'interface ont d'abord été représentés dans une représentation directement utilisable par simple copie dans la mémoire écran (image bitmap)

■ Aujourd'hui: utilisation des capacités d'affichage évoluées des ordinateurs pour:

- Autoriser le graphisme (vectoriel) dans les interfaces
- Autoriser l'animation dans les interfaces
- Améliorer l'esthétique des interfaces
- Imaginer de nouveaux interacteurs

page 156



Bénéfices du vectoriel

■ Indépendance de la résolution

- Facilite le passage à des plus hautes résolutions
- Facilite le passage à des basses résolutions
 - Anti-aliasing sur le texte
 - (Adaptation -notamment du texte- à la taille de l'écran)

■ Nouvelles possibilités

- Animation
- Transparences
- Effets

■ Problème: performances

- Tire parti des accélérateurs graphiques

page 157





Séparer interface et traitement

■ Séparer le traitement de l'interface

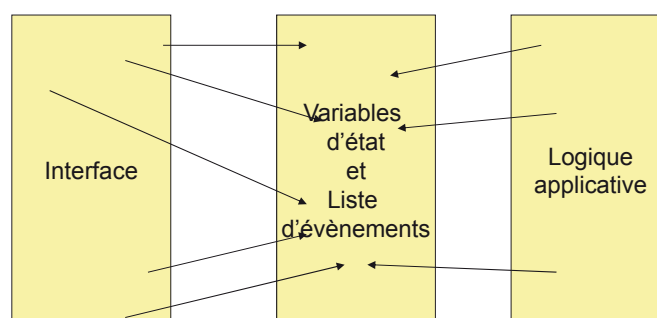
■ Permet

- De mettre au point le traitement indépendamment de l'interface
- De mettre au point l'interface indépendamment du traitement
- De disposer de plusieurs interfaces (expert, novice...)
- De définir des éléments d'interface réutilisables
 - Et reconnaissables par les utilisateurs

page 158



Exemple de séparation



Asynchronisme entre l'application et son interface

page 159





IHM et Navigateur

- **Utilisation d'une interface Web**
 - Soit à l'intérieur d'une application en intégrant un objet navigateur dans l'application
 - Soit avec une application navigateur en déléguant le traitement
 - Applet, soap/web services...
- **Permet de décrire l'interface avec un fichier séparé de l'application et simple à éditer (HTML/SVG)**
- **Délégation du traitement à des web services**
 - Web services = traitement déclenchés par une requête HTTP
 - Et réponse de la même façon
- **Permet de présenter des composants de traitement partout sur le réseau**
- **Permet d'utiliser ces composants par des interfaces partout sur le réseau**

page 160



Concept de client léger

- **Application client/serveur**
- **La partie client ne fait que de la présentation, généralement via un navigateur**
 - Éventuels traitements faits au niveau du client
 - vérifications sur les paramètres entrés
 - traitements locaux de données pour synchroniser plusieurs présentations d'une même donnée
- **Des traitements sont faits sur le serveur**
- **Choix crucial de la répartition des traitements entre le client et le serveur**

page 161





Niveaux de description de l'interface

- **Son rôle: l'interface abstraite**
 - Exemple: choix dans une liste
- **Sa présentation**
 - Exemple: liste déroulante de type menu avec texte noir sur fond bleu
 - Exemple: liste audio
- **Sa relation avec du code exécutable**
 - Le code que l'interface sert à déclencher

page 162



Lien entre interface et code

- **La description de l'interface fait référence à du code**
 - Par exemple, en associant un appel de fonction à des événements que peut déclencher l'interface
 - Onclick, onmouseup,...
- **La description du code fait référence à l'interface**
- **Un document intermédiaire ou des portions spécifiques du code définissent la relation entre éléments d'interface et éléments de code**

page 163



Exemples : code 'DOM Events'

```
<script type="application/ecmascript" >
  function doSomething(evt) {
    ...
  }
</script>
<text onclick="doSomething(evt)" >Hello World!</text>
```

Lien dans l'interface

Lien dans le code

```
<script type="application/ecmascript" >
  function doSomething(evt) {
    ...
  }
  e = document.getElementById('T');
  e.addEventListener('click', doSomething, false);
</script>
<text id="T" >Hello World!</text>
```

page 164

TELECOM ParisTech

Interfaces: langages déclaratifs

- UIML
- XUL
 - Firefox
- XAML
 - Microsoft Windows
- Android Layouts
- UIDL
 - <http://www.uidl.net/>
- ...

page 165

TELECOM ParisTech



Langages de description d'interface

■ Origine

- Fichiers de ressources isolés de l'application
- Solution primaire pour modifier l'interface sans modifier le code applicatif et vice versa
 - Des données isolées du code proprement dit décrivent des éléments d'interface (menus, boîtes de dialogue...)
 - Un moteur d'interprétation de ces données est intégré à l'application (en général via des fonctions du système) pour présenter l'interface et passer les résultats des interactions au code applicatif via des événements
- Tant que les liaisons entre l'interface et le code sont stables, chaque partie peut être modifiée indépendamment de l'autre

page 166



User Interface Markup Language

- Initié en 1997, puis normalisé par OASIS
- But:
 - décrire les interfaces une fois
 - Les décliner sur diverses plateformes à l'aide d'...
- Outils de transformation de la description d'interface en interface réelle
- Semble utilisé par des applications, mais nombreux développements autour d'UIML semblent abandonnés
- Site: <http://www.uiml.org/>

page 167





XML User interface Language

■ Outil de la fondation Mozilla

- Sert, par exemple, pour l'interface du navigateur Firefox
- Site de référence: www.mozilla.org/projects/xul/
- Spécifications: www.mozilla.org/projects/xul/xul.html
- Site utile: xulfr.org , www.xulplanet.com

■ Interface décrite en 4 parties

- Le contenu
- L'apparence (style/skin)
- Le comportement
- La localisation

page 168



Contenu

■ Ensemble de boîtes rectangulaires


- Contenant d'autres boîtes ou des éléments d'interface (boutons, liste...)
- Avec des informations de mise en page relative
 - Tel éléments est à droite de tel autre...
- Avec des informations relatives au redimensionnement de la fenêtre dans laquelle elles sont

■ Exemple

```
<?xml version="1.0"?>
<window xmlns="...">
  <box align="center">
    <button label="Clique ici" onclick="alert('Bonjour le monde !');" />
  </box>
</window>
```

page 169







Gadgets

- **Mini-application**
 - Conçue pour rendre un service précis, limité
 - Apparaît dans la barre latérale du bureau de Vista, sur le bureau
 - Code HTML+script+CSS
 - Packagé dans un fichier .gadget


page 170

W3C Widgets

- <http://www.w3.org/TR/widgets/>
- Octobre 2007
- « Widgets are a class of client-side web application for displaying and/or updating local or remote data, packaged in a way to allow a single download and installation on a client machine or device. Examples include clocks, stock tickers, news casters, games and weather forecasters. This specification, when combined with other dependent specifications, defines a software solution for Widgets, including... »

page 171





Rich Internet Application

■ Applications

- dont l'interface est un navigateur
- Mais dont la partie client n'est pas limitée à de la présentation (a contrario des clients légers)

■ Avantages

- Pas besoin d'installer le logiciel
- Disponibilité sur toute machine avec un accès Internet

page 172



Technologies pour les RIA

■ Javascript

- AJAX: Asynchronous Javascript and XML
 - Exemple: GMail, GoogleMaps, Flickr
- Frameworks Javascript (>500)
 - Yahoo User Interface Library
 - DOJO
 - Scriptaculous
 - Google Web Toolkit

■ Flash

- Actionscript, Flex
- Lecteur Flash accessible de 98% des navigateurs

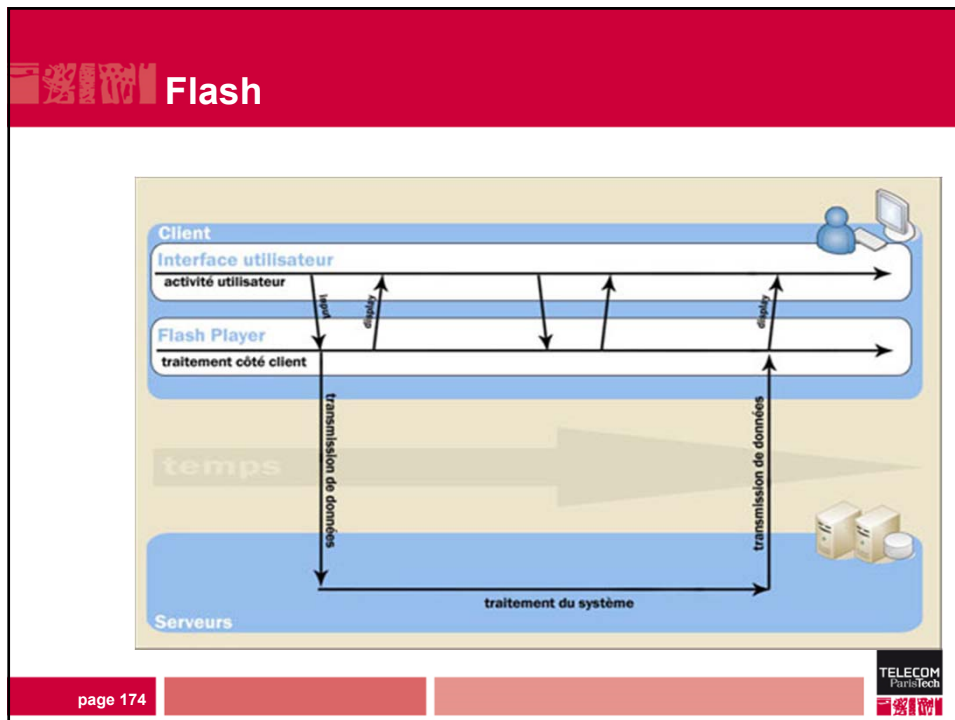
■ Java

- Applets
 - Nécessitent la présence d'une machine virtuelle Java
- Alternative Java Web Start

■ JavaFX (Sun), Silverlight (Microsoft)

page 173






page 174

AIR: RIA à l'envers?

- **Macromedia a introduit une sorte de machine virtuelle dans les navigateurs avec son player Flash**
 - D'abord un player d'animation vectorielle
 - + des hyperliens
 - + un peu de script
 - Est peu à peu devenu un environnement de développement d'applications client/serveur
- **Adobe/Macromedia cherche aujourd'hui à faire sortir sa machine virtuelle du navigateur pour en faire un support de développement d'applications multiplateforme**

TELECOM ParisTech logo is visible in the bottom right corner.


page 175



Applications basées Web

- **XUL runner**
 - Pour des applications de bureau avec une interface web basée sur les outils de Firefox
- **PhoneGap/Cordova**
 - Pour des applications mobiles avec une interface web

page 176



Web et Multimedia mobile





Pourquoi s'intéresser aux mobiles?

- **2,5 billion de possesseurs de téléphones mobiles**
 - 1 billion de PC
- **GSM couvre 80% de la population mondiale**
- **En Europe, 28% des possesseurs de téléphone ont utilisés Internet sur leur mobile... (source IPSOS)**
- **... mais 10 à 15% suivant les pays utilisent Internet sur leur mobile (source BBC)**

chiffres 2011



Interfaces pour mobiles

- **Intense concurrence**
- **Plusieurs stratégies**
 - Navigateur(+SVG) comme moteur d'interface (Opera)
 - Java+SVG (Sun)
 - Technologie basée sur MPEG-4 Laser/SVG (Streamezzo)
 - Flash Lite (Adobe Macromedia)
 - XAML (Microsoft)





La standardisation

- **W3C**

- L'organisme de normalisation du Web

- **OMA: Open Mobile Alliance**

- **3GPP**

- **MPEG**

- ...

=> secteur en évolution rapide

=> efforts de standardisation

=> intense concurrence

=> 'standards' concurrents



W3C et mobiles

- **XHTML, XHTML Basic, XHTML Mobile Profile**

- **CSS**

- **SVG Mobile**

- **Mobile Web Initiative**

- <http://www.w3.org/Mobile/>





Outils du W3C

- <http://www.w3.org/Mobile/>
- **MobileOK Validator**
 - <http://validator.w3.org/mobile/>
- **Séminaires, cours, tutoriaux**
 - Best practices
- **Ensemble de tests et exemples (test suites)**
- **Description des terminaux**
 - À la façon de WURFL
 - <http://www.w3.org/TR/DDR-Simple-API/>



Web Mobile Best Practices

- **Recommandation du W3C**
 - 60 pratiques recommandées
 - Profil minimum de terminaux ayant accès au Web Mobile
 - Donne des indications sur la capacité d'un site à être lu
 - Permet à des développeurs de contenus de se forger une compétence claire



'Best practices' (2)

- Réduire le nombre de ressources externes
- Limiter la taille des pages
- Scroller dans une seule direction
- Contenu principal en début de page
- Mise en page: pas de table, pas de frames, pas d'espacement graphique
- Éviter les images trop grandes
- Page compréhensible sans couleur
- Bon contraste entre le contenu et le fond
- Texte de remplacement des contenus non textuels
- Expliciter les informations telles que la taille des images
- Contenu non dépendant des objets embarqués et des scripts



Outils pour le Web Mobile

- **Outil de validation issu de la Mobile Web Initiative**
 - Mobile Web Best Practices Checker
 - <http://validator.w3.org/mobile/>
 - Wiki: <http://www.w3.org/2005/MWI/BPWG/techs/>
- **Mobi**
 - <http://ready.mobi/index.html>





Approche anti-'One Web'

- **Principe: le contenu doit être adapté au medium**

- =>

- Ne pas servir un site Web sur un mobile
- Ne pas servir un site pour mobile sur le Web

- **Une référence**

- "Global Authoring Practices for the Mobile Web"
- <http://www.passani.it/gap/>



Pas le plus petit dénominateur commun

- **XHTML Mobile Profile**

- Définit par l'OMA
- Sous ensemble de XHTML
- Sur ensemble de XHTML Basic
 - Ajoute WAPCSS et l'utilisation de styles
 - Ajoute des possibilités graphiques
- Tutoriels
 - XHTMLTUT - XHTML Mobile Profile 1.0 Tutorials
 - XHTML MP tutorial at Developers Home: <http://www.developershome.com/wap/xhtmlmp/>
 - XHTML MP tutorial at Openwave: http://developer.openwave.com/dvl/support/documentation/guides_and_references/xhtml_tutorial/index.htm





Diversité des terminaux

- Plus de 6000 terminaux recensés par le projet WURFL en 2010
- Les opérateurs de téléphonie considèrent qu'ils ont environ 300 terminaux différents chez leurs clients
- 700 terminaux différents dans le monde à un moment donné
- D'autres référentiels de terminaux existent
 - DeviceAtlas,...



WURFL

- <http://wurfl.sourceforge.net/>
- Un projet coopératif de constitution et mise à jour d'un répertoire des terminaux diffusés dans le monde
- Gratuit
- Support de nombreux langages
- Repose sur la chaîne user_agent usuellement envoyée par un navigateur dans son entête
 - NokiaN70-1/4.0548.2.0.3 Series60/2.8 Profile/MIDP-2.0 Configuration/CLDC-1.1
 - SAMSUNG-SGH-T108+/1.0*MzUwOTUxMDg1MDI5OTA4 UP/4.1.19k





Caractéristiques significatives (1)

- **Taille de l'écran**
- **Taille des images**
- **Nombre de couleurs**
 - NOIR ET BLANC, 4096, 65535, 16 M
- **Formats d'images reconnus et préférés**
 - Jpeg, Ping...
- **Fréquence d'échantillonnage sonore**
- **Formats de sons reconnus**
- **Nombre de lignes et de colonnes de texte**
 - Ex écran 128x140 et texte sur 14 lignes de 8 caractères, soit des caractères de 9 par 17 pixels au maximum



Caractéristiques significatives (2)

- **Capacité de traitement (CPU)**
- **Accélérations matérielles**
 - 3D, 2D, Cryptage, ...
- **Capacité mémoire**
- **Limites sur la taille des objets chargés**
- **Support des cookie**





Caractéristiques significatives (3)

■ Support de Java

- Version de J2ME (Java Micro-Edition)
- APIs Java supportées
- Support de l'accélération hard par Java

■ Support du Web (navigateur intégré)

- Type de langage de balise supporté et préféré (XHTML Basic...)



Taille de l'écran: analyse de WURFL (1)

En juin 2006, analyse de

■ 74 paires (largeur, hauteur) différentes

- De 64x98 à 770x300 (et aussi 0x0 => pas d'écran)
- Largeur de 64 à 770
- Hauteur de 24 à 480
- Exemples avec largeur 101
 - Hauteurs: 33, 54, 64, 67, 111
 - Possibilité de confondre 101x64 et 101x67
 - Sur ce principe, on peut se ramener à une trentaine de cas





Taille de l'écran: réduction à +/-12% (2)

h='72' w='26'/>
 h='84' w='48'/>
 h='96' w='44'/>
 h='96' w='24'/>
 h='96' w='52'/>
 h='101' w='67'/>
 h='101' w='33'/>
 h='110' w='72'/>
 h='101' w='80'/>
 h='120' w='108'/>
 h='112' w='84'/>
 h='128' w='121'/>
 h='131' w='128'/>
 h='123' w='69'/>
 h='125' w='96'/>
 h='136' w='120'/>
 h='150' w='100'/>
 h='160' w='128'/>

h='160' w='140'/>
 h='160' w='160'/>
 h='176' w='144'/>
 h='180' w='160'/>
 h='181' w='116'/>
 h='198' w='176'/>
 h='208' w='176'/>
 h='208' w='104'/>
 h='240' w='160'/>
 h='256' w='240'/>
 h='256' w='118'/>
 h='320' w='240'/>
 h='352' w='324'/>
 h='360' w='120'/>
 h='480' w='320'/>
 h='490' w='165'/>
 h='640' w='200'/>
 h='770' w='300'/>



Alternatives à WURFL

- <http://www.openddr.org/>
- <http://deviceatlas.com/>
- <http://51degrees.mobi/>
- <http://detectmobilebrowsers.mobi/>
- <http://code.google.com/p/php-mobile-detect/>
- <http://detectmobilebrowsers.com/>
- ...





Outils de validation

■ Les émulateurs

■ Problèmes:

- Qualité de l'émulation
- En particulier, maîtrise des performances réelles



Applications spécifiques

■ Exploitation de la parole

- Messages, commandes vocales...

■ Exploitation de la localisation

- Contenu change en fonction de la position géographique du terminal

■ Exploitation de la mobilité

- Ex: jeu en réseau en liaison avec le monde réel





Aborder un projet

- **Parc de terminaux visés**
- **Design de site avec la perspective**
 - De la consistance thématique sur mobiles
 - La disponibilité de contenu pour mobiles
- **Prévoir des tests sur de vrais terminaux**
 - Pas seulement des émulateurs



Particularités de la conception pour mobiles

- **Petit écran**
- **Possibilité en entrée limitées**
- **Réseau lent et couteux**
- **Diversité des terminaux**
- **Terminal très personnel**
- **Toujours allumé**
- **... mais pb d'alimentation**
- **Toujours connecté**
- **Appelable**
- **Terminaux de large diffusion (=> peu chers)**





Caractéristiques importantes

- **Petit écran**
- **Manipulations d'une seule main**
- **Habituellement**
 - une navigation suivant 4 directions
 - Pas de clavier complet
- **Faible bande passante**
- **Latence élevée**
- **Connexion incertaine**
- **Attention limitée**
- **Mouvement/mobilité**



Pourquoi la diversité des terminaux va perdurer?

- **Les fabricants cherchent à différencier leurs produits**
- **Les usages sont très variés**
 - Variabilité des besoins d'autonomie d'énergie
 - Variabilité des besoins d'affichage
 - ...
- **Les opérateurs veulent des terminaux spécifiques**
- **Avancées rapides des technologies et innovation**
- **18 mois pour 'sortir' un nouveau terminal**
- **Il est courant que les mises à jour de logiciel ne soient pas effectuées**





Réduire la complexité

- **Identifier un contexte restreint**
 - Ex: un pays
- **Passer par des abstractions**
 - Ex: choix dans une liste
 - Ex: <http://www.w3.org/2009/dap/>
- **Limiter les ambitions**
 - Ex: recommandations du W3C



Identifier un contexte restreint

- **Un pays**
 - On passe de 700 à 300 terminaux
- **Une entreprise**
 - => une flotte identifiable de terminaux
- **Les possesseurs d'un type précis de terminal**
- **Les possesseurs de terminaux ayant au moins certaines caractéristiques**
 - Ex: écran de taille supérieure à 120x120



103



Adaptation à l'utilisateur

Langue, goûts,
handicaps,
préférences de
présentation,
préférences en
terme de
consommation
de débit...



Source ptimousse34.skyblo



Page 206 - titre - 12/01/2006



Adaptation au terminal

- Plus de 5000 terminaux décrits par le projet WURFL



Page 207 - titre - 12/01/2006

Adaptation au contexte

- **Réseau d'accès**
 - Bande passante
 - Taux d'erreur
 - Présence d'une voie de retour
- **Environnement lumineux et sonore**
- **Autonomie électrique du client**
- ...

Page 208 - titre - 12/01/2006



Explosion combinatoire

Langues

x tailles d'écran

x débit

x configuration soft

x choix d'adaptation préférés par l'utilisateur

...

= trop de situations d'adaptation pour une réponse prédéfinie

Page 209 - titre - 12/01/2006



Adaptation

■ Au niveau du serveur

- Se servir de l'identification du terminal client pour fournir un contenu qui lui correspond
 - Soit par simple choix dans un ensemble de contenus prédéfinis
 - Soit en générant au vol un contenu adapté

■ Au niveau du client

- Modification de l'affichage au niveau du player du client
 - Par exemple, d'après la feuille de style CCS

■ Entre les deux (proxy d'adaptation)

- Exemple: adaptation sur une passerelle entre les serveurs de contenu et une flotte de terminaux définie



Adaptation de documents multimédia

De nombreuses possibilités

■ Adaptation de chaque média

- Transcodage
- Changement de modalité
- Transformation

Mais aussi

■ Adaptation de l'organisation spatiale et temporelle de ces médias

- Ex: répartir une page sur plusieurs pages
- Ex: afficher des images en séquence sur un petit écran tandis qu'on les affiche simultanément sur un grand
- Ex: supprimer un texte qui commente une image si l'image a du être supprimée

■ Adaptation du mode d'acheminement des médias

- Possible distribution sur divers terminaux





Responsive Design

Mot clé des méthodes d'adaptation les plus pratiquées sur le web

S'appuient principalement sur le CSS3

définition des caractéristiques d'affichage de classes de contenus en fonction de paramètres identifiés par le navigateur: taille d'écran, langue...



Liens

- <http://www.adobe.com/devnet/svg.html>
- <http://svg-wow.org/>
- <http://thegraphicalweb.com>
- <http://tvmjong.free.fr/INKSCAPE/MANUAL/html/>
- <https://developer.mozilla.org/en-US/docs/SVG>
- <http://www.webplatform.org/>
- <http://carto.net/>
- OpenStreetMap <http://openstreetmap.fr/>
- OpenClipArt <http://openclipart.org/>
- <http://svgopen.org>
- <http://pilatinfo.org/index.html>
- <http://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html>
- <http://srufaculty.sru.edu/david.dailey/svg/>
- <http://demos.hacks.mozilla.org>
- <http://ie.microsoft.com/testdrive/>
- <https://hacks.mozilla.org/demos/> (<https://hacks.mozilla.org/category/svg/>)
- <http://www.svgbasics.com>
- <http://www.siteduzero.com/tutoriel-3-14858-le-svg.html>

Inkscape
Guide to a Vector
Drawing Program
Fourth Edition



Tamijang Ltd.



WebPlatform.org

